

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, информатики и информационных технологий  
Кафедра информационно-коммуникационных технологий в образовании

# ПОДКАСТ-ТЕРМИНАЛ ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ

*Выпускная квалификационная работа  
бакалавра по направлению подготовки  
09.03.02 – Информационные системы и технологии*

Исполнитель: студент группы ИС-41  
Института математики, информатики и ИТ  
Хомутинин С.С.

Руководитель: к.п.н., доцент кафедры ИКТО  
Сардак Л.В.

Работа допущена к защите  
« \_\_\_\_ » \_\_\_\_\_ 2016 г.  
Зав. кафедрой \_\_\_\_\_

Екатеринбург – 2016

## Реферат

Хомутинин С.С. ПОДКАСТ-ТЕРМИНАЛ ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ, выпускная квалификационная работа: 50 стр., рис. 25, табл. 2, библи. 23 назв., приложений 1.

*Ключевые слова:* подкастинг, подкаст-терминал, web-разработка, Python, Django, информационная система.

*Объект разработки:* подкаст-терминал образовательного учреждения.

*Цель разработки:* спроектировать подкаст-терминал образовательного учреждения (УрГПУ) и реализовать локальный прототип системы средствами инструментов web-разработки.

В работе произведен анализ технологий реализации подкаст-терминала, выявлена специфика подкаст-терминала учебного назначения. Разработано техническое задание на создание прототипа информационной системы «Подкаст-терминал образовательного учреждения». На основе технического задания, информационная система была реализована. Теоретически описаны этапы разработки полноценной информационной системы.

Локальный прототип информационной системы «Подкаст-терминал образовательного учреждения» был реализован средствами языка программирования Python и фреймворка Django. Данная система имеет возможность модульного расширения функционала.

## Оглавление

<b>ВВЕДЕНИЕ.....</b>	<b>4</b>
<b>ГЛАВА 1. ПОНЯТИЕ И ТЕХНОЛОГИЯ РЕАЛИЗАЦИИ ПОДКАСТ-ТЕРМИНАЛА .....</b>	<b>6</b>
1.1 ПОНЯТИЕ ПОДКАСТ-ТЕРМИНАЛА .....	6
1.2 ТЕХНОЛОГИИ РЕАЛИЗАЦИИ ПОДКАСТ-ТЕРМИНАЛА .....	10
1.3 ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПОДКАСТ-ТЕРМИНАЛА ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ .....	15
<b>ГЛАВА 2. РАЗРАБОТКА ПОДКАСТ-ТЕРМИНАЛА ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ .....</b>	<b>23</b>
2.1 МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «ПОДКАСТ-ТЕРМИНАЛ ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ» .....	23
2.2 РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ .....	30
2.3 РЕЗУЛЬТАТЫ АПРОБАЦИИ, ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ .....	47
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>49</b>
<b>СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....</b>	<b>50</b>
<b>ПРИЛОЖЕНИЯ .....</b>	<b>52</b>
Приложение 1. ....	52

## Введение

Согласно пункту 7.3.4 Федерального государственного образовательного стандарта высшего образования (ФГОС ВО) приказом от 12 марта 2015 г. №219 (в ред. Приказа Минобрнауки России от 09.09.2015 № 999), «обучающимся должен быть обеспечен доступ (удаленный доступ), в том числе в случае применения электронного обучения, дистанционных образовательных технологий, к современным профессиональным базам данных и информационным справочным системам, состав которых определяется в рабочих программах дисциплин (модулей) и подлежит ежегодному обновлению» [9]. Одним из направлений развития дистанционных образовательных технологий в формате электронных образовательных ресурсов является развитие системы трансляции учебного видео. Передаваемый видеопоток, зачастую, имеет достаточно большой информационный объем, что является препятствием для использования подобного представления информации при малой пропускной способности сети. Такие проблемы могут возникнуть при использовании беспроводного подключения по Wi-Fi, если точка доступа находится далеко от приемника сигнала, или перегружена большим количеством абонентов. Мобильные 3g и 4g сети тоже не являются стабильным подключением, да и затраты за использованный трафик ложатся на обучающегося. Одним из решений может служить использование технологии подкастинга, то есть разбиение видео потока на небольшие по объему и времени трансляции фрагменты, которые размещаются на специальном сервере. Собственный подкаст-терминал в образовательном учреждении позволит учащимся получать учебные материалы дистанционно – можно слушать или смотреть подкасты находясь дома или в общественном транспорте в любое желаемое для учащегося время. Доступ к подкастам может осуществляться как с персонального компьютера через браузер, так и с помощью мобильного приложения.

В ходе разработки методики применения технологии подкастинга в обучении А.А. Володиным и А.А. Володиным был сделан вывод о том, что «...технология подкастинга, как инновационное информационно-коммуникационное средство обучения, занимает важное место в системе современных средств обучения. Технология подкастинга, за счет присущих ей дидактических свойств и функций, обладает рядом достоинств, которые способны сделать процесс обучения более эффективным и, в конечном итоге, способствовать качественному достижению поставленных дидактических целей.» [17]. Данный вывод подтверждает актуальность применения технологий подкастинга в образовательном процессе высшего учебного заведения.

Объект разработки: подкаст-терминал образовательного учреждения.

Цель разработки: спроектировать подкаст-терминал образовательного учреждения (УрГПУ) и реализовать локальный прототип системы средствами инструментов web-разработки.

Для достижения поставленной цели выделим следующие задачи:

1. Произвести анализ технологий реализации подкаст-терминала, выявить специфику подкаст-терминала образовательного учреждения.
2. Проанализировать и обосновать выбор аппаратно-программной платформы, определить структуру, объекты web-интерфейса подкаст-терминала образовательного учреждения.
3. В соответствии с техническим заданием смоделировать, спроектировать и реализовать локальную версию прототипа подкаст-терминала образовательного учреждения.
4. Подготовить техническую и сопроводительную документацию, провести апробацию.

# **Глава 1. Понятие и технология реализации подкаст-терминала**

## **1.1 Понятие подкаст-терминала**

На сегодняшний день технологии подкастинга являются распространенным способом передачи информации в сети Интернет. Данные технологии могут быть развлекательного, информационного, научно-популярного или образовательного характера.

Термин «Подкаст» является синтезом двух слов – iPod и broadcast – повсеместное вещание, широковещение [19]. Портативный медиапроигрыватель iPod компании Apple являлся наиболее популярным средством прослушивания подкастов. Определение смежных терминов «Подкаст» и «Подкастинг» на сайте компании Apple звучит так: «Подкастинг – это разновидность информационного вещания в Интернете. Как правило, подкасты – это оригинальные аудио- или видеозаписи, но также бывают записи теле- или радиопередач, лекций, выступлений и других событий» [14].

Первые упоминания подкастинга связаны с 2004 годом [20], а уже в 2005 году по мнению редакции Нового Оксфордского американского словаря термин «Подкаст» стал «словом года». Значение данного термина определялось как «цифровая запись радиотрансляции или аналогичной программы, которую можно скачать через Интернет для прослушивания в персональном аудиоплеере» [22].

В данной работе под понятием «подкаст» будут пониматься цифровые аудио- или видеозаписи, чаще серии записей, имеющие определенную тематику и распространяемые через интернет. Исходя из определения подкаста, подкаст-терминал будет представлять собой сайт, организующий работу с подкастами и выполняющий различные смежные функции (подписка на обновления, сбор обратной связи и т.д.). Подкасты размещаются на сайте авторами в соответствующих им тематических категориях, после чего с помощью RSS-

рассылки аудио или видео файлы автоматически поступают на персональный компьютер тем, кто подписан на эту рассылку.

Подкасты, как правило, приближены к формату радиопередач. Это могут быть обзоры новостей, рассуждения одного или нескольких человек на определенную тематику, интервью или обучающие лекции.

Подкаст может представляться в следующих формах:

- аудиофайл. Обыкновенная звуковая запись без сопровождающего видеоряда;
- аудиофайл с фоторядом или слайдкаст. Аудио-подкаст может сопровождаться какими-либо иллюстрациями. Это могут быть фотографии, схемы или таблицы;
- видеофайл. Может представлять собой запись с экрана, съемку на камеру или захват с веб-камеры автора подкаста с соответствующим аудиорядом.

Любой подкаст по желанию автора может сопровождаться текстовой версией, а также содержать любые необходимые дополнительные материалы. Подкаст может воспроизводиться в потоковом режиме, либо может быть скачан как с веб-версии, так и с мобильной.

Для авторов подкастов веб-версия подкаст-терминала должна обеспечивать следующие задачи:

- размещение подкастов на сайте с помощью интуитивного интерфейса в любой форме представления;
- распределение подкастов по необходимым категориям, добавление ключевых слов (тегов);
- сбор обратной информации – статистика просмотров и т.д.

Для пользователей задачи следующие:

- удобная навигация по различным разделам подкастов;
- онлайн воспроизведение и скачивание подкастов;

- управление RSS-подписками на авторов, какой-либо раздел, либо серию подкастов.

В мобильной версии, как правило, возможность добавления подкастов авторами отсутствует.

Рассмотрим примеры подкастов в сервисе iTunes [17].

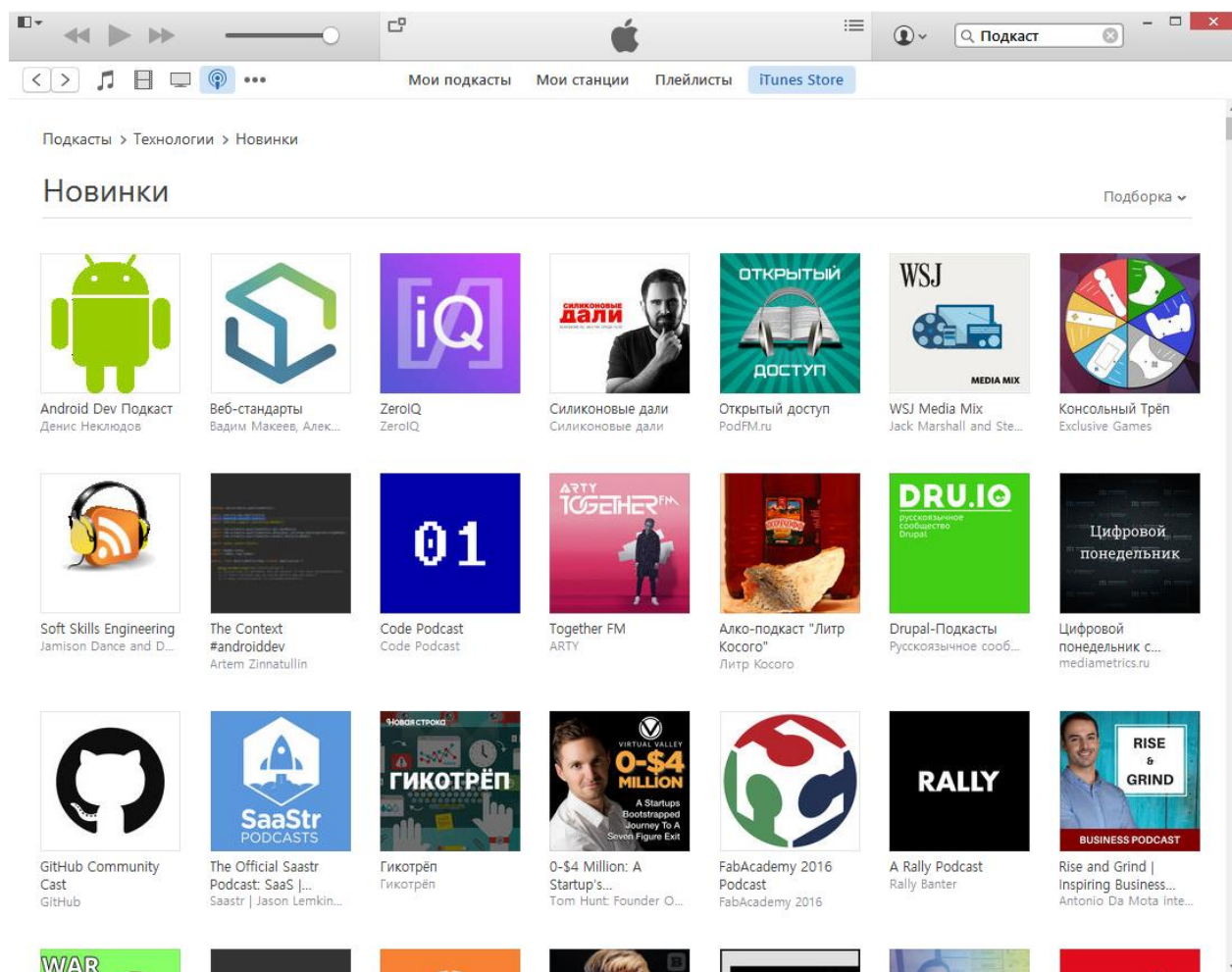


Рисунок 1. Сервис iTunes. Список новых подкастов из раздела "Технологии"

В разделе «Технологии» можно найти подкасты различных форматов. Например, в подкасте «Android Dev Подкаст» рассматриваются различные тематики, связанные с разработкой приложений под операционную систему Android. Серия подкастов обладает последовательной подачей материала, предназначенного для уже имеющих опыт разработки таких приложений.



Другого формата серия подкастов «Веб-стандарты». В ней рассматриваются различные новости веб-разработки за неделю несколькими авторами – подкастерами.

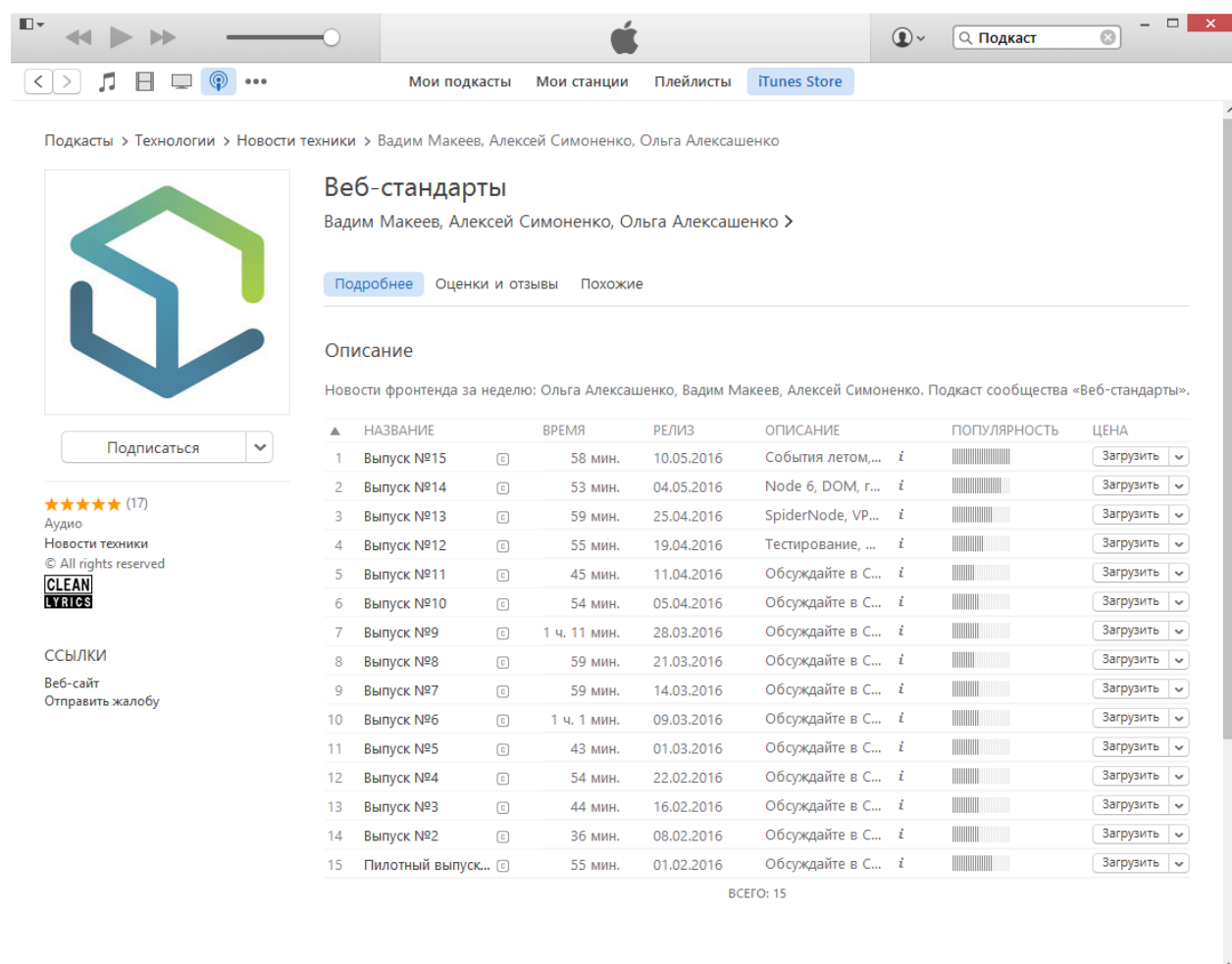


Рисунок 2. Серия подкастов "Веб-стандарты"

Серия подкастов «Силиконовые дали» имеет формат радио-шоу, где в каждом выпуске ведущие и приглашенный гость обсуждают какую-либо тематику, связанную с интернет-технологиями.

В серии подкастов «Открытый доступ» собраны различные аудиокниги и аудиостатьи из открытых источников, в том числе и переводы на русский язык зарубежной литературы.

Подобные форматы подкастов характерны не только для раздела «Технологии», но и для любой другой категории.

## 1.2 Технологии реализации подкаст-терминала

Подкаст-терминал обычно представляет собой сайт или веб-приложение, организующий работу с подкастами. Функционал таких сайтов может различаться, и для реализаций различного функционала могут потребоваться различные технологии, отвечающие своим целям и задачам. Примером технологий для быстрой реализации систем с простым функционалом являются CMS – системы управления содержанием [7]. Преобладающее большинство сайтов не обладающих широким или специфическим функционалом создаются с помощью таких систем. Процесс создания предельно прост и зачастую даже не требует знания языков программирования, или требует знать лишь самые основы. В настоящее время популярны такие CMS, как WordPress, Joomla, Drupal [6].

Многие CMS разрабатывались с целью автоматизировать или максимально упростить разработку типовых, похожих друг на друга сайтов – блогов, интернет-магазинов и т.д. Это приводит к тому, что одни CMS подходят для какой-либо конкретной задачи лучше, чем другие. Данное утверждение хорошо продемонстрировано примерами CMS Joomla и WordPress в анализирующей их статье Д.С. Абдуразаковой, В.Е. Демашиной, Д.В. Лучанинова: «CMS WordPress часто выбирают в качестве платформы для блога. <...> CMS Joomla часто применяется для создания крупных и сложных по функционалу проектов. Кроме того, Joomla используется в качестве CMS для разработки интернет-магазина. <...> CMS Joomla более приближена к порталным системам, нежели CMS WordPress» [1].

Другой способ – самостоятельная разработка сайта без использования CMS-систем. Такой подход актуален, например, для разработки клиент-серверных веб-приложений. Разработку любого сайта можно разделить на две большие части, непосредственно связанные друг с другом – разработка того, что видит конечный пользователь, или клиентская часть (front-end), и разработка серверной части (back-end) [2].

На сегодняшний день front-end разработка представляет собой работу с совокупностью технологий, каждая из которых является обязательной и выполняет определенную функцию – это языки программирования HTML, CSS, JavaScript. HTML является языком разметки, с помощью которого можно задать структуру сайта – расположение блоков, таблиц, каких-либо списков и т.д. CSS отвечает за оформление элементов сайта, а JavaScript служит для добавления динамики – проверки заполнения форм, всплывающих окон и т.д. Для данных технологий существуют различные фреймворки (англ. *framework* — каркас, структура) [10], реализующие различные подходы к написанию кода и предоставляющие некоторые базовые возможности, от которых можно отталкиваться при разработке front-end части сайта.

Для back-end разработки существует широкий выбор языков программирования и фреймворков к ним. На сегодняшний день наиболее популярными языками для написания серверной части являются:

- PHP (Фреймворки: Symfony, Laravel, Kohana);
- Python (Django);
- Ruby (Ruby on Rails);
- языки .NET (ASP.NET);
- Java (Spring MVC, Vaadin, JSF);
- Perl (Catalyst).

Большинство языков программирования не обязывают использовать какой-либо определенный веб-сервер. Целесообразно использовать один из следующих веб-серверов:

- Apache;
- Nginx;
- Lighttpd.

ASP.NET поддерживается далеко не всеми веб-серверами. Самым популярным веб-сервером, который поддерживает ASP.NET является IIS.

Немаловажной частью веб-приложения является система управления базами данных (СУБД). Популярными на сегодняшний день СУБД являются:

- MySQL;
- PostgreSQL;
- MS SQL Server;
- Oracle Database.

Выбор технологий реализации подкаст-терминала образовательного учреждения должен производиться исходя из специфики конечного продукта. Даже среди подкаст-терминалов общего назначения часто встречаются различия в функционале. Подкаст-терминал образовательной направленности будет обладать своим, уникальным функционалом, нетипичным для обычных подкаст-терминалов. Специфику подкаст-терминала образовательного учреждения демонстрирует следующая таблица [12, с.140]:

*Таблица 1.*  
*Отличительные особенности обычного подкаст-терминала и подкаст-терминала в образовательном учреждении*

<b>Обычный подкаст-терминал</b>	<b>Подкаст-терминал в образовательном учреждении</b>
Простая структура пользователей. Авторы выкладывают свои подкасты в определенных категориях, слушатели подписаны на каких-либо авторов или категории.	Более сложная структура с разными уровнями доступа. Обучающиеся получают доступ только к тем подкастам, которые входят в их образовательную программу.
Пользователь подписывается на тех авторов, которых он желает слушать.	Обучающийся автоматически подписан на конкретные предметы, которые ему необходимо изучать. За предметами может быть закреплено несколько преподавателей, и наоборот – преподаватель может вести несколько предметов.
Пользователь может общаться с другими слушателями конкретных подкастов, вступать в дискуссии, оставлять отзывы и пожелания.	Обучающемуся может быть представлено какое-либо задание на проверку усвоения материала. Это может быть тест, или письменное задание.

Многие авторские подкаст-терминалы представляют собой блог, созданный с помощью каких-либо CMS [8]. Данные системы вполне могут быть полезны для создания простых подкаст-терминалов, однако разработка учебного подкаст-терминала с помощью CMS может быть затруднительна из-за сложной структуры пользователей и разнообразия функционала.

Однако реализация подкаст-терминала образовательного учреждения с помощью систем управления содержимым может иметь несколько недостатков. Во-первых, большинство CMS не дают полной свободы действий при реализации проекта, заставляя подгонять разработку под шаблоны, свойственные многим CMS. Во-вторых, работа с CMS не обеспечит углубленное изучение технологий создания как клиентской части, так и серверной. Специфика подкаст-терминала образовательного учреждения такова, что реализация ее особенностей средствами CMS может быть сильно затруднена, поэтому использование данных систем при разработке подобного проекта не рекомендуется.

Самостоятельная разработка полноценного веб-приложения не накладывает никаких ограничений на проект, поэтому имеется возможность реализации всех особенностей подкаст-терминала образовательного учреждения. Кроме этого, подобный подход к разработке позволяет углубиться в изучение спектра используемых технологий.

Для разработки клиентской части сайта стоит использовать современные возможности языков HTML5, CSS3, JavaScript.

HTML5 – последняя версия стандарта HTML, рекомендованная к использованию с 2014 года. О достоинствах и причинах использовать преимущества HTML5 можно судить по высказыванию Р. Никсон: «...с новыми, высокоэффективными свойствами HTML5, такими как холсты, аудио, видео и геолокация, можно придать вашим веб-страницам более высокую динамичность, интерактивность и мультимедийность» [5, с. 41].

CSS3 – последняя законченная версия стандарта каскадных таблиц стилей. CSS3 не является единым стандартом, так как он был разбит на несколько независимых модулей, стандартизация которых разрабатывается отдельно. По многим модулям уже сегодня ведется разработка CSS4, но говорить о их применении вместо CSS3 пока еще рано.

JavaScript, как и предыдущие технологии, не имеют прямых альтернатив, поэтому его использование обязательно. Данный язык может выполнять функцию серверного языка вместо каких-либо отдельных технологий. Некоторые разработчики находят это удобным т.к. отпадает необходимость изучения большого спектра различных технологий. Но наиболее распространенная область применения данного языка – пользовательские скрипты в браузере. По причине выбора другой серверной технологии, JavaScript будет использоваться только в качестве инструмента написания пользовательских скриптов.

При выборе серверных технологий обычно отталкиваются от языка программирования, на котором хотелось бы разрабатывать веб-приложение. Для каждого языка программирования есть перечень популярных фреймворков, реализующих различные задачи. Некоторые фреймворки настоятельно рекомендуют использовать совокупность каких-либо определенных технологий. Для серверной части подкаст-терминала будет использоваться Python с фреймворком Django. В качестве веб-сервера будет использоваться nginx. В роли СУБД оптимально использовать PostgreSQL. Все эти технологии являются негласным стандартом при разработке средствами Django и используются преобладающим большинством разработчиков веб-приложений на языке Python.

Python на сегодняшний день является одним из самых распространенных языков программирования. Он прост в освоении, во многих школах основы программирования преподаются именно на этом языке. В то же время Python обладает высокой функциональностью, поддержкой объектно-ориентированного программирования и богатой стандартной библиотекой. Данный язык программирования может использоваться в различных областях

деятельности. Так как Python является востребованным и многофункциональным, целесообразно его использование в разработке веб-приложения с целью изучения технологий.

Фреймворк Django для языка Python используется специально для разработки веб-приложений. Это не единственный фреймворк Python для данной цели, но он является наиболее функциональным и распространенным, а также обладает отличной документацией и большим сообществом, что, несомненно, помогает при разработке. Данный фреймворк реализует шаблон проектирования MVC (Model-view-controller), рассмотренный В.Л. Чургеевым [13].

PostgreSQL является стандартной рекомендуемой СУБД для Django. Это свободная объектно-реляционная СУБД, являющаяся одной из самых популярных СУБД в мире [21].

В реализации подкаст-терминала образовательного учреждения могут использоваться любые актуальные на сегодняшний день серверные технологии – завершённый проект для конечного пользователя будет иметь одинаковый функционал вне зависимости от выбранных технологий.

Итого, перечень используемых технологий выглядит следующим образом:

- клиентская часть – HTML5, CSS3, JavaScript;
- серверная часть – Python с фреймворком Django;
- СУБД – PostgreSQL.

### **1.3 Техническое задание на разработку подкаст-терминала образовательного учреждения**

#### **1. Общие сведения**

##### **1.1. Название организации – заказчика**

Уральский государственный педагогический университет (УрГПУ).

##### **1.2. Название продукта-разработки**

Подкаст-терминал образовательного учреждения.

### **1.3. Назначение продукта**

Данный продукт предназначен для организации системы закрытого подкастинга в образовательном учреждении.

## **2. Характеристика области применения продукта**

### **2.1. Процессы и структуры, в которых предполагается использование продукта разработки**

Оперативное обеспечение образовательного процесса мультимедийным контентом с использованием веб-технологий в Уральском государственном педагогическом университете.

### **2.2. Характеристики персонала (количество, квалификация, степень готовности)**

Подкаст-терминал образовательного учреждения имеет следующие типы пользователей:

- администратор;
- преподаватель;
- студент.

Администратор портала должен уметь работать с административной частью подкаст-терминала и обладать знаниями о его структуре.

Преподаватели должны обладать знаниями о создании мультимедийного контента средствами специального программного обеспечения, уметь размещать созданный контент с помощью интерфейса подкаст-терминала.

Студенты должны обладать основами работы в сети интернет.

## **3. Требования к продукту разработки**

### **3.1. Требования к продукту в целом**

Подкаст-терминал образовательного учреждения предоставляет преподавателям возможность размещать учебные мультимедийные материалы в виде подкастов в категориях, соответствующих учебным дисциплинам.

Студенты получают возможность прослушивания, просмотра и скачивания тематических подкастов, входящих в программу их подготовки.



### **3.2.Аппаратные требования**

Для реализации серверной части портала требуется хостинг с дисковой квотой не менее 200 Гб.

Требования клиентской части будут различны для каждого типа пользователей.

Студенты должны обладать персональным компьютером или мобильным устройством с выходом в сеть Интернет со скоростью, достаточной для воспроизведения или скачивания мультимедийного контента портала.

Администраторы и преподаватели должны обладать персональным компьютером с выходом в сеть Интернет, а также специальными устройствами, необходимыми для записи подкастов:

- микрофон;
- веб-камера.

### **3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.)**

Требования к программному обеспечению хостинга:

- UNIX-подобная операционная система;
- веб-сервер Apache с возможностью использования модуля `mod_wsgi`;
- база данных PostgreSQL 8.2 и выше;
- Python 3.2 и выше;
- наличие зарегистрированного и корректно настроенного домена.

Студенты должны иметь возможность пользоваться порталом на любой операционной системе с помощью любого современного веб-браузера.

Администратор и преподаватели должны обладать специальным программным обеспечением для записи подкастов.

### **3.4.Указание программного обеспечения, используемого для реализации**

Интерпретатор языка Python – CPython.

Фреймворк для языка Python – Django.

Разработка серверной и клиентской части – среда разработки PyCharm (Версии 4 и выше). По необходимости – Notepad++.

### **3.5. Особенности реализации серверной и клиентской частей**

Клиентская часть не предусматривает возможности работы с подкастами, кроме их просмотра. Такая возможность имеется только в интерфейсе администратора.

Серверная часть должна представлять собой максимально независимые друг от друга модули. Должна быть возможность расширения существующего проекта дополнительными модулями.

### **3.6. Форматы входных и выходных данных**

Входные данные в виде медиафайлов преобразуются в структурированные подкасты. Поддерживаемые форматы файлов:

- .ogg;
- .mp4;
- .webm.

### **3.7. Источник данных и порядок их ввода в систему (программу), порядок вывода, хранения**

Администратор средствами интерфейса администратора по спискам дисциплин и учебным планам задает структуру, добавляя в систему направления подготовки, дисциплины и занятия. По необходимости, добавляет подкасты в какие-либо дисциплины.

Преподаватели с помощью ограниченного интерфейса администратора вводят в систему медиафайлы и текстовое описание к ним. Как единая сущность введенные данные размещаются и хранятся на сервере.

### **3.8. Порядок взаимодействия с другими системами, возможности обмена информацией**

Не предусмотрено.

### **3.9. Меры защиты информации**

Вход в систему осуществляется после авторизации пользователя. Контент закрытый, т.е. доступен только авторизованным пользователям. Доступ к редактированию структуры подкаст-терминала и любых подкастов есть только у пользователя, имеющего права администратора. У пользователей с правами преподавателя имеется доступ к редактированию подкастов только в определенных дисциплинах.

## **4. Требования к пользовательскому интерфейсу**

### **4.1. Общая характеристика пользовательского интерфейса**

В системе представлено три типа пользователей, обладающих собственными вариациями интерфейсов.

Администратор имеет доступ к полноценному интерфейсу с широкими возможностями по редактированию информации, содержащейся в системе. Эти возможности включают в себя:

- работу с пользователями. Добавление и редактирование пользователя, занесение первичной информации о нем, распределение прав доступа и принадлежности к группам;
- работу с группами. Добавление и редактирование групп и распределение прав доступа для каждой конкретной группы пользователей;
- работу со структурой. Добавление и редактирование направлений подготовки, учебных дисциплин и занятий;
- работу с подкастами. Добавление и редактирование подкастов в определенных занятиях.

Преподаватель имеет доступ к интерфейсу администратора, но ограничен в возможностях редактирования информации теми дисциплинами, которые за ним закреплены. В этих дисциплинах преподаватель имеет возможность осуществлять работу с подкастами.

Интерфейс студента не предполагает доступа к интерфейсу администратора. Студент получает доступ только к просмотру определенных его направ-

лением подготовки дисциплин, занятий и подкастов. Администратор и преподаватель имеет доступ к просмотру подкастов любых направлений подготовки и дисциплин.

#### 4.2. Размещение информации на экране. Дизайн экрана

В шапке располагается логотип образовательного учреждения и название веб-сервиса. Возможно размещение какой-либо информации об образовательном учреждении. В правой части шапки содержится информация о текущем пользователе и кнопка выхода. Основная информация занимает среднюю и правую часть экрана. Слева расположено древо дисциплин и принадлежащих им занятий для более удобной навигации по ним.



Рисунок 3. Макет страницы с выбором дисциплин и занятий

Страница просмотра подкаста над основной информацией содержит названия дисциплин, занятий и самого подкаста. Окно воспроизведения медиафайла расположено в средней части, а в правой части – выбор других подкастов из текущего занятия. Снизу располагается описание текущего подкаста.

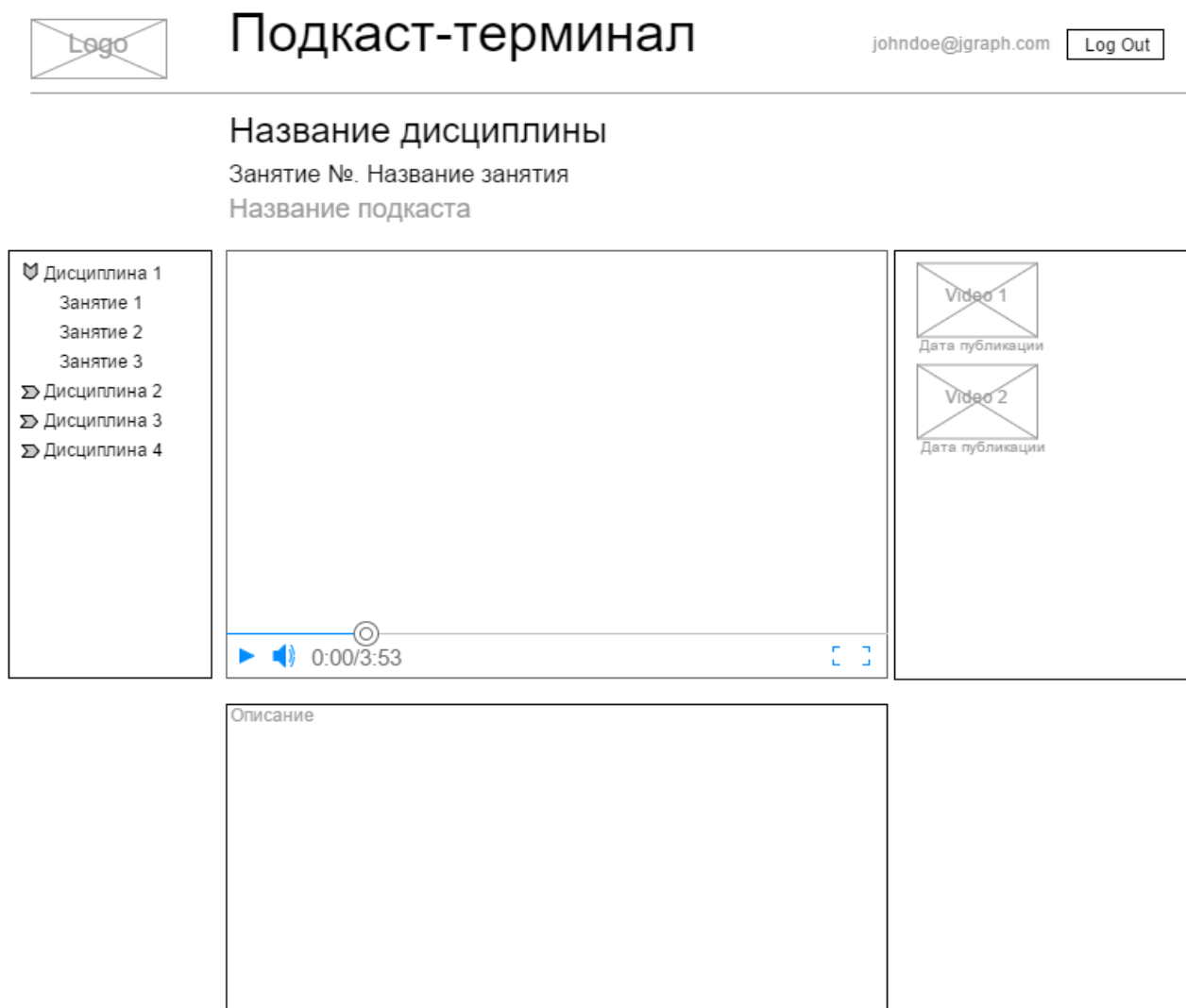
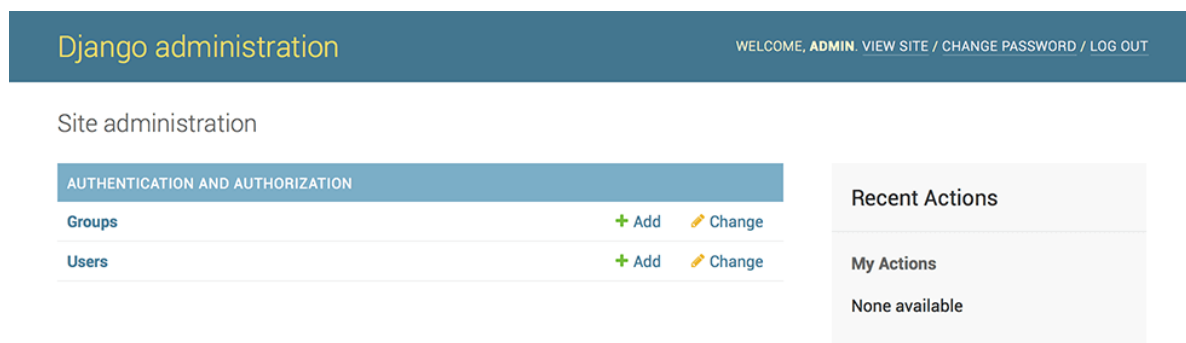


Рисунок 4. Макет страницы просмотра подкаста

Интерфейс администратора представлен стандартными средствами фреймворка Django и имеет собственную структуру. В шапке расположены ссылки на просмотр самого сайта, изменение пароля и выход. В правой части – последние действия различных пользователей в административной части сайта. По середине представлены стандартные модели и работа с ними. К стандартным моделям относятся группы и пользователи, входящие в состав подключае-

мого модуля аутентификации и авторизации. По ходу разработки должны быть подключены к данному интерфейсу другие модели, с которыми будет производиться работа. Например, модель подкастов.



*Рисунок 5. Страница стандартного интерфейса администратора Django*

## **5. Перечень сопроводительной документации**

Руководство пользователя.

## **6. Порядок сдачи-приемки продукта**

В соответствии с графиком выполнения ВКР.

## **Глава 2. Разработка подкаст-терминала образовательного учреждения**

### **2.1 Моделирование информационной системы «подкаст-терминал образовательного учреждения»**

Процесс разработки системы «Подкаст-терминал образовательного учреждения» можно разделить на три этапа жизненного цикла:

- проектирование и разработка;
- тестирование;
- эксплуатация.

В рамках моделирования информационной системы, была разработана модель деятельности процесса разработки данной системы. Модель включает в себя три этапа жизненного цикла и следующих пользователей, принимающих участие в каких-либо этапах процесса разработки:

- разработчик;
- администратор;
- преподаватель;
- студент.

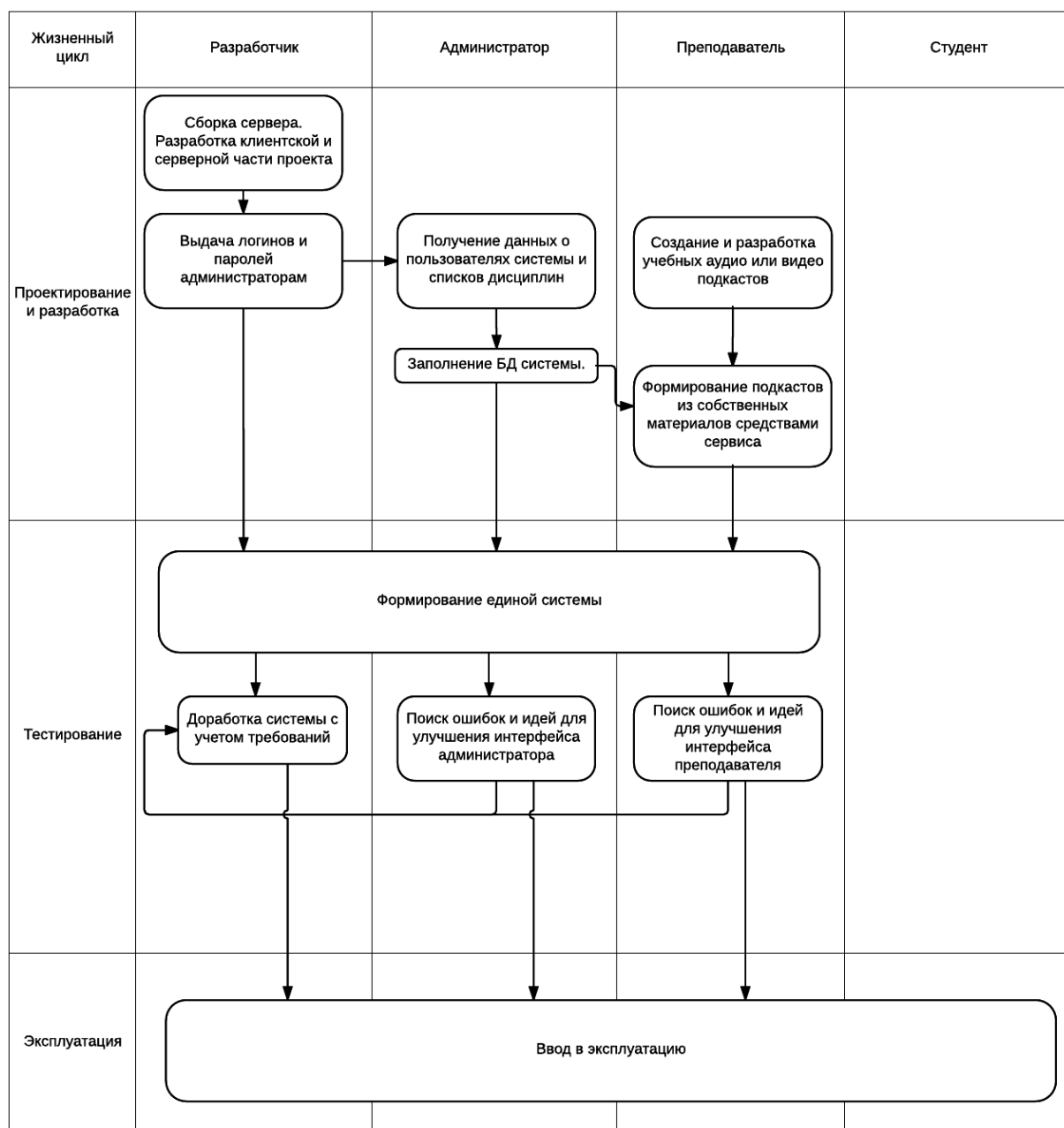


Рисунок 6. Модель деятельности процесса разработки информационной системы «Подкаст-терминал образовательного учреждения»

На этапе проектирования и разработки системы разработчик и преподаватели выполняют независимые друг от друга задачи параллельно.

Разработчик занимается проектированием информационной системы. Производит разработку программной и аппаратной части системы. Преподаватели занимаются созданием будущей базы учебных видеоматериалов и структурируют их по различным учебным дисциплинам и занятиям.



После создания функционирующего прототипа разработчик создает одного или нескольких пользователей с правами администратора и выдает их логины и пароли непосредственно будущим администраторам системы. Администраторы получают данные о будущих пользователях системы (преподавателях и студентах) и списки дисциплин соответственно федеральному государственному образовательному стандарту высшего образования. Имея доступ к разрабатываемой системе, администраторы занимаются заполнением базы пользователей и структуры учебных дисциплин, а также выдачей прав пользователям и распределением их по группам. После заполнения базы пользователей (в том числе преподавателей), преподаватели получают доступ к системе и формируют на основе своих учебных видеоматериалов подкасты средствами информационной системы.

На этапе тестирования деятельность разработчика, преподавателей и администраторов формируется в единую систему. Разработчиком производится исправление недоработок и улучшение существующего функционала на основании обратной связи с преподавателями и администраторами. Студенты в данном этапе жизненного цикла, как и в предыдущем, участия не принимают.

После тестирования системы и подтверждения ее работоспособности, начинается ее ввод в эксплуатацию. Этап эксплуатации является завершающим этапом жизненного цикла системы. Студенты получают доступ к системе и используют ее в образовательном процессе. Разработчик, администраторы и преподаватели обеспечивают поддержку информационной системы, дополняя или дорабатывая ее по мере необходимости.

Кроме модели деятельности процесса разработки системы, важной частью моделирования является функциональная модель готовой информационной системы.



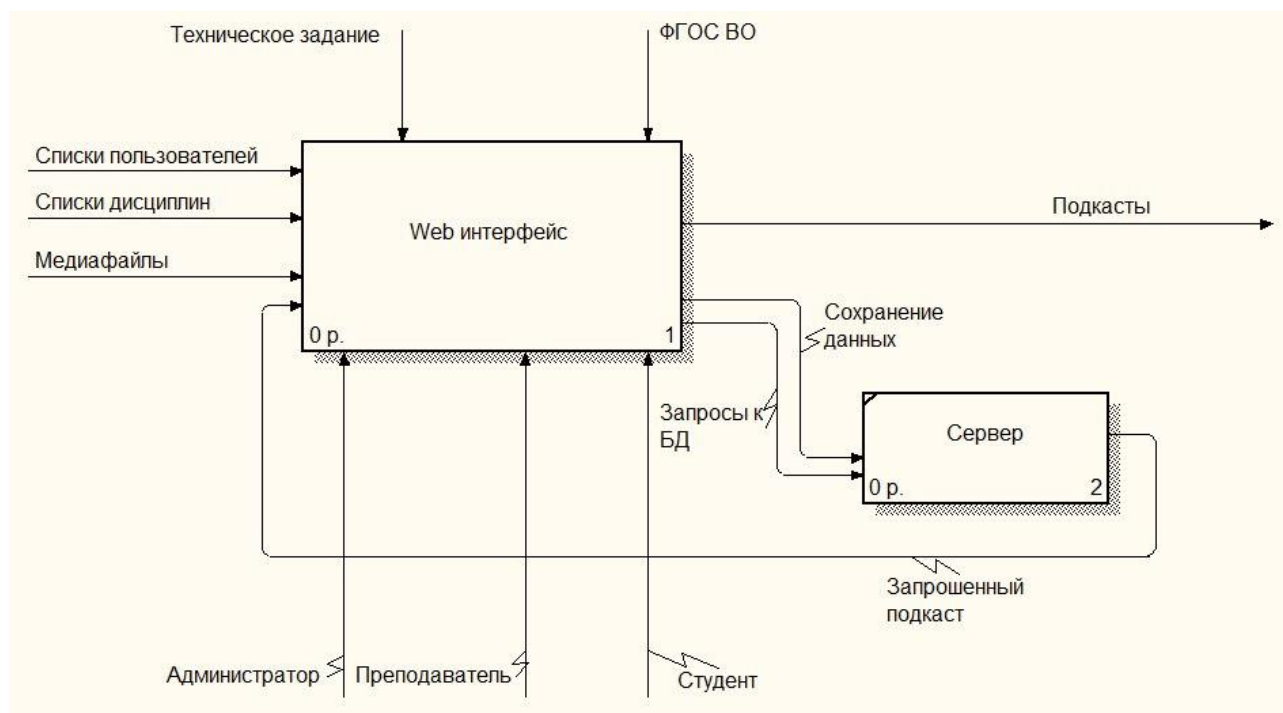
Рисунок 7. Функциональная модель информационной системы «Подкаст-терминал образовательного учреждения»

Работать с информационной системой могут три типа пользователей – администратор портала, преподаватели учебного заведения и студенты. Стоит отметить, что после запуска информационной системы в эксплуатацию происходит совмещение функционала разработчика и администратора.

Для организации работоспособности портала необходимо наполнить его контентом, таким как списки пользователей и дисциплин. Списки дисциплин включают в себя информацию об учебных дисциплинах согласно федеральному государственному образовательному стандарту высшего образования. Списки пользователей содержат информацию о преподавателях и учебных дисциплинах, которые за ними закреплены, а также информацию о студентах, их направлениях подготовки и принадлежности к различным учебным группам.

В систему вводятся медиафайлы, служащие основой учебного контента в информационной системе. В результате функционирования информационной системы медиафайлы преобразуются в подкасты и подаются на выход.

Второй уровень функциональной модели показывает связи между двумя модулями – web интерфейсом и сервером баз данных.



*Рисунок 8. Второй уровень функциональной модели информационной системы «Подкаст-терминал образовательного учреждения»*

Основной функционал обеспечивает web интерфейс. Все пользователи напрямую не взаимодействуют с серверной частью, за это отвечает web интерфейс, который при необходимости отправляет запросы к серверу баз данных. Медиафайлы средствами web интерфейса сохраняются на сервере баз данных в структурированной единой сущности – подкасте. В зависимости от запросов, поступающих на сервер через web интерфейс, возвращается какая-либо информация. Например, дисциплины, пользователи, подкасты.

Портал разделен на два интерфейса. Преподаватель не имеет собственного интерфейса и работает с ограниченным интерфейсом администратора.

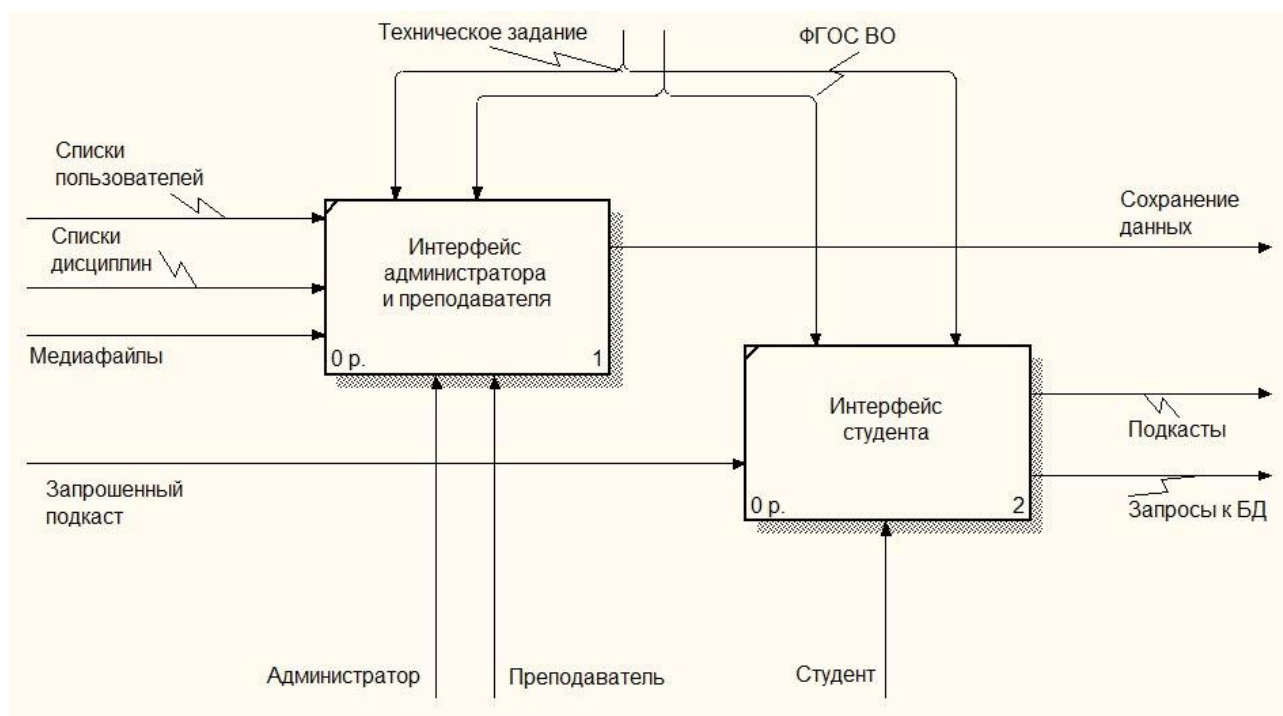


Рисунок 9. Детализация web интерфейса функциональной модели информационной системы «Подкаст-терминал образовательного учреждения»

Все данные, полученные системой, обязательно проходят через интерфейс администратора. Сохранение какой-либо информации также возможно только средствами данного интерфейса. Интерфейс студента предполагает лишь возможность отправлять запрос на получение информации из сервера баз данных и отображать ее. Доступ к интерфейсам возможен лишь одноименным типам пользователей – администратор и преподаватели имеют доступ к интерфейсу администратора, а студент к интерфейсу студента.

Детализация интерфейса администратора показывает различия доступа к нему преподавателей и администратора, а также распределение входящих материалов по различным модулям интерфейса администратора.

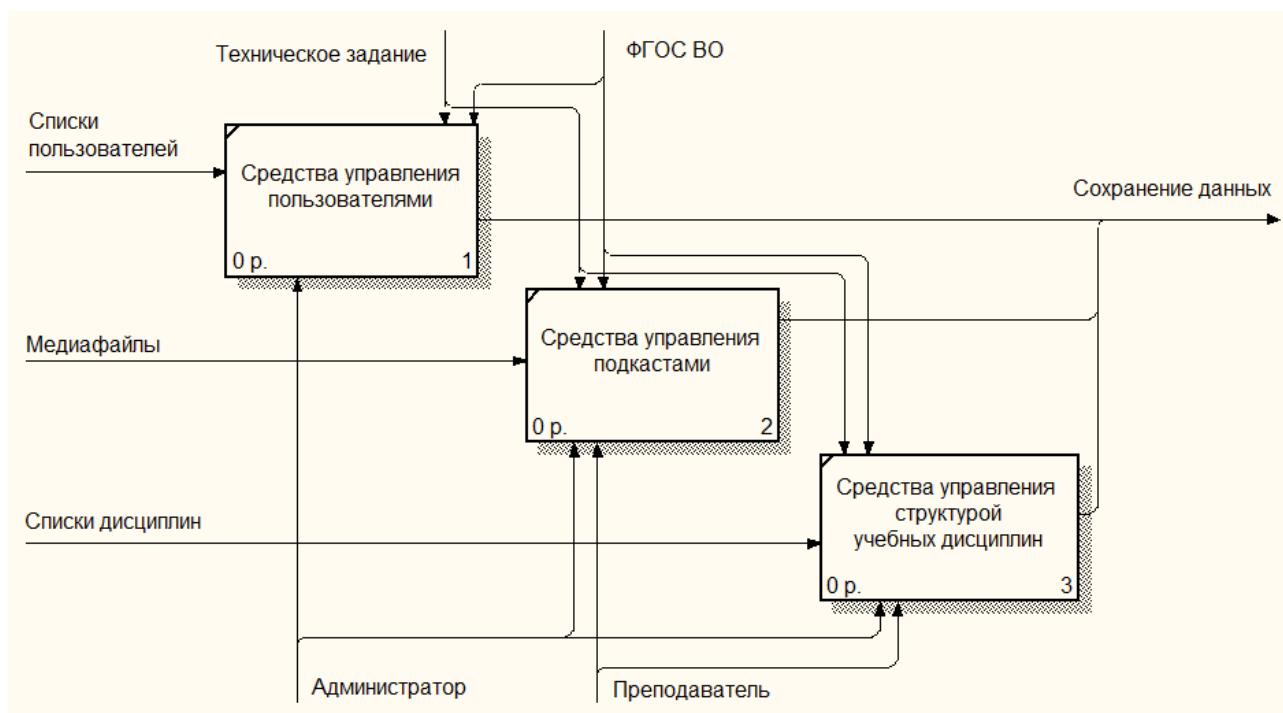


Рисунок 10. Детализация интерфейса администратора и преподавателя функциональной модели информационной системы «Подкаст-терминал образовательного учреждения»

Администратор имеет полноценный доступ ко всем средствам управления. По спискам пользователей администратор добавляет и распределяет по группам преподавателей и студентов в информационной системе. Списки дисциплин администратор преобразует в структуру учебных дисциплин системы. По необходимости, администратор добавляет в какие-либо дисциплины подкасты на основании входящих медиафайлов. Преподаватель получает доступ к средству управления подкастами и структурой учебных дисциплин. Но, в отличие от администратора, его доступ к этим средствам не является полноценным. Преподаватель может управлять подкастами и структурой занятий только в определенных его правами дисциплинах. Каждое из средств управления напрямую работает с сервером баз данных и обеспечивает изменение и сохранение какой-либо информации.

Таким образом, на этапе проектирования информационной системы были выделены структурные элементы подкаст-терминала и связи между ними, а также определен перечень пользователей с соответствующим им функционалом в системе. Проектирование структурных элементов позволяет определить ход разработки серверной части и реализации интерфейсов.

## **2.2 Реализация информационной системы**

Процесс реализации информационной системы можно разбить на несколько последовательных этапов:

1. Локальная установка программного обеспечения;
2. Разработка серверной части информационной системы. Реализация интерфейсов;
3. Развертывание прототипа проекта на сервере;
4. Модификация информационной системы;
5. Создание сопроводительной документации.

Данный параграф содержит теоретическое описание перечисленных этапов и описание функционала разработанной информационной системы.

### **2.2.1 Локальная установка программного обеспечения**

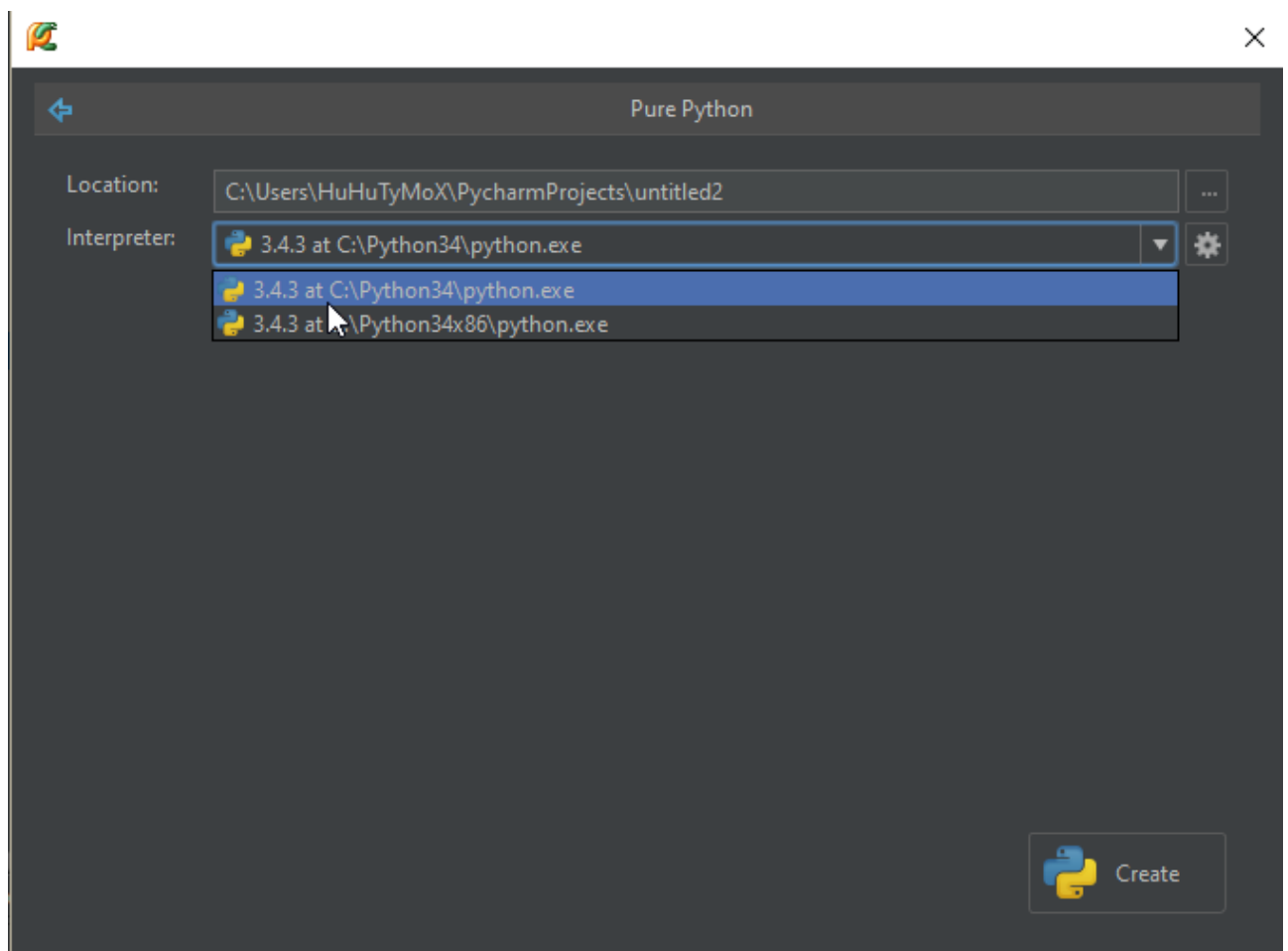
Программное обеспечение устанавливается в соответствии с техническим заданием. Все используемое программное обеспечение распространяется на бесплатной основе.

Интерпретатор языка Python – CPython расположен на официальном сайте языка программирования Python [23] в разделе Download. Рекомендуется версия 3.x и выше. Установка представляет собой стандартную установку программного обеспечения.

Среда разработки PyCharm требует установленного интерпретатора языка программирования Python. Данная среда разработки не обязательна т.к. можно успешно разрабатывать и запускать приложения Python и без нее, но PyCharm

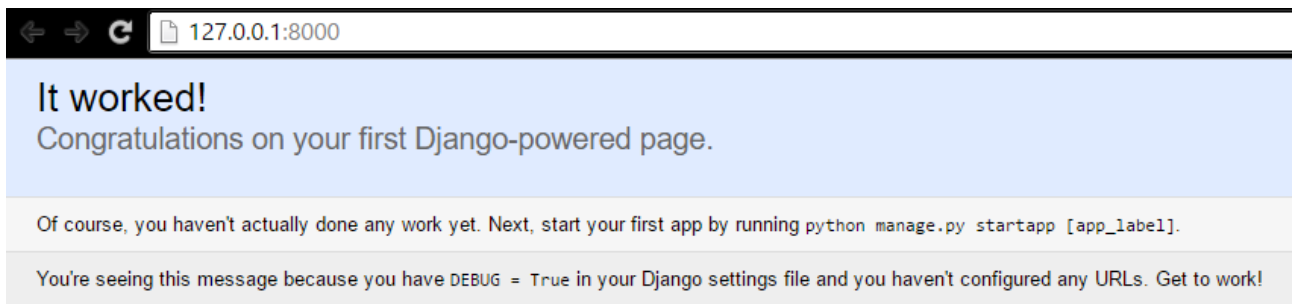
сильно упрощает работу с данными технологиями. Скачивание среды разработки возможно на официальном сайте компании-разработчика JetBrains [18]. PyCharm существует в двух редакциях – Professional и Community. Выбрать можно любую.

После установки и запуска среды разработки при создании приложения Python в поле Interpreter необходимо выбрать ранее установленный интерпретатор Python.



*Рисунок 11. Выбор интерпретатора в среде разработки PyCharm при создании нового проекта*

Среда разработки PyCharm позволяет создавать проекты средствами фреймворка Django и берет вопросы его установки на себя. После создания проекта чтобы убедиться, что все работает, следует нажать Shift+F10 и перейти в веб-браузере по адресу <http://127.0.0.1:8000/>.



*Рисунок 12. Проверка работоспособности проекта Django*

Фреймворк Django обладает интегрированным локальным веб-сервером для более удобной разработки, что позволяет уменьшить перечень используемого программного обеспечения.

### 2.2.2 Разработка серверной части информационной системы. Реализация интерфейсов

Процесс разработки средствами Django представляет собой создание независимых модулей, направленных на выполнение какой-либо функции в информационной системе. По умолчанию при создании проекта имеется только стандартный модуль этого проекта, отвечающий за общие настройки, html-шаблоны, связь с другими модулями и т.д. Все настройки располагаются в файле **settings.py**. Например, в этом файле описывается подключение к базе данных, если оно необходимо:

```
1. DATABASES = {
2.     'default': {
3.         'ENGINE': 'django.db.backends.postgresql_psycopg2',
4.         'NAME': 'podcasts',
5.         'USER': 'postgres',
6.         'PASSWORD': 'pass',
7.     }
```

Файл **urls.py** отвечает за конфигурацию URL-адресов страниц. При обращении к какому-либо URL-адресу, будет вызываться соответствующая ему функция, либо подключаться другой файл конфигурации.



Данная конфигурация запускает функцию `views.login_view` при открытии страницы по адресу `/accounts/login/`:

```
1. url(r'accounts/login/', views.login_view),
```

Данная конфигурация не запускает никакую функцию, но передает управление другому файлу конфигурации, который отвечает за модуль интерфейса администратора:

```
1. url(r'^admin/', include(admin.site.urls)),
```

Функции, запускаемые при обращении к какому-либо адресу описаны в файле **views.py** и в среде разработки Django называются представлениями (англ. View- вид, взгляд). Каждое представление оформляется как функция, возвращающая какой-либо html-шаблон, или перенаправляющая на другую страницу. Например, небольшая функция выхода из информационной системы перенаправляет на главную страницу:

```
1. def logout_view(request):
2.     logout(request)
3.     return redirect('/')
```

Файлы **urls.py** и **views.py** функционируют обычно вместе. Данные файлы могут располагаться в любом отдельном модуле, если он отвечает за какие-либо URL-адреса. Например, URL-адреса, по которым расположен какой-либо подкаст, конфигурируются данными файлами в рамках модуля `podcasts` и только в нем. Базовый модуль, получив запрос на подобную ссылку, перенаправляет его в модуль подкастов.

Файл **models.py** является опциональным, как и многие другие файлы, но выполняет важную функцию. В нем описываются модели, аналогичные сущности в базе данных, используемой в проекте. Большое преимущество фреймворка Django в том, что он позволяет использовать все средства работы с базой данных не обращаясь к ней напрямую и не используя язык СУБД – например, SQL. Все модели описываются на языке Python и средствами системы миграций фреймворка Django транслируются в запросы к базе данных.

Следующим образом выглядит описание модели подкаста:

```
1. class Podcast(models.Model):
2.     podcast_name = models.CharField(max_length=30)
3.     podcast_exercise = models.ForeignKey('structure.Exercise')
4.     podcast_text = models.CharField(max_length=200)
5.     pub_date = models.DateTimeField('date published')
6.     link = models.FileField(max_length=200)
```

Любой модуль, имеющий какие-либо модели в своей структуре, можно подключить к интерфейсу администратора для более удобного управления этими моделями с помощью файла **admin.py** с подобным содержимым:

```
1. class PodcastAdmin(admin.ModelAdmin):
2.     fields = ['podcast_name', 'podcast_exercise', 'podcast_text',
3.             'pub_date', 'link']
4.     list_display = ('podcast_name', 'podcast_exercise', 'pub_date')
5.     admin.site.register(Podcast, PodcastAdmin)
```

Многие модули обладают необходимостью их отображения на html-странице, которую видит пользователь. Существуют следующие рекомендации для генерации подобных страниц в фреймворке Django:

- создание главного шаблона (макета, заготовки) сайта в основном модуле информационной системы;
- разделение шаблона на определенные блоки, заполнение которых будет зависеть от других модулей;

Такой подход предотвращает дублирование кода, а также способствует его структурированию и пониманию. Шаблоны хранятся в директории `templates` внутри директории модуля. Статическая информация вроде каскадных таблиц стилей и скриптов JavaScript располагается в папке `static`.

Основной шаблон обычно содержит ту информацию, которая не меняется в зависимости от открытой страницы. Например, шапка сайта или какие-то отдельные части.

Те области сайта, в которых может располагаться какой-либо контент, обычно выносятся в блоки. Выглядит это следующим образом:

```
1.         <div class="col-md-4">
2.             {% block content %}
3.             {% endblock %}
4.         </div>
```

Данный синтаксис вставок в html-код специфичен для Django. По умолчанию блок content пуст, но в зависимости от открытой страницы он будет содержать какую-либо информацию. Для этого необходимо в модуле, информацию которого нужно отобразить, создать файл с шаблоном, содержимым которого будет описание нужного блока, и подключить основной шаблон следующей командой в начале файла:

```
1.  {% extends "Portal/base.html" %}
```

В рамках разработки локального прототипа информационной системы «Подкаст-терминал образовательного учреждения» можно выделить следующие моменты, результат реализации которых будет описан далее в разделе описания информационной системы:

- базовая настройка информационной системы. Подключение необходимых модулей, базы данных, установка стандартных директорий;
- проектирование и разработка используемых в системе модулей. Создание необходимых моделей, подключение интерфейса администратора к ним;
- конфигурация URL-адресов и разработка представлений к ним;
- создание базового и наследующих его шаблонов;
- работа с интерфейсом администратора. Заполнение системы.

### 2.2.3 Развертывание прототипа проекта на сервере

Основной платформой развертывания веб-приложения на языке программирования Python является WSGI – стандарт взаимодействия с веб-сервером, например, Apache.

Для корректной работы веб-сервера необходимо произвести базовую конфигурацию, которая предполагает собой редактирование файла `httpd.conf` веб-сервера.

```
1. WSGIScriptAlias / /path/to/mysite.com/mysite/wsgi.py
2. WSGIPythonPath /path/to/mysite.com
3.
4. <Directory /path/to/mysite.com/mysite>
5. <Files wsgi.py>
6. Require all granted
7. </Files>
8. </Directory>
```

Значение `WSGIScriptAlias` указывает местоположение ваших приложений или проектов. Вторым значением указывается расположение файла «WSGI» – как правило, он находится в корне проекта.

Значения `WSGIPythonPath` и `<Directory>` также необходимо настроить согласно примеру.

Следует удостовериться, что файл `wsgi.py` существует. По умолчанию, он появляется при создании проекта. Если по какой-либо причине его нет, необходимо создать файл самостоятельно.

#### 2.2.4 Модификация информационной системы

В связи с тем, что проект Django предполагает собой разбиение на модули, имеется возможность модификации информационной системы с целью привести в нее дополнительный функционал. Для подкаст-терминала образовательного учреждения может быть полезен следующий функционал:

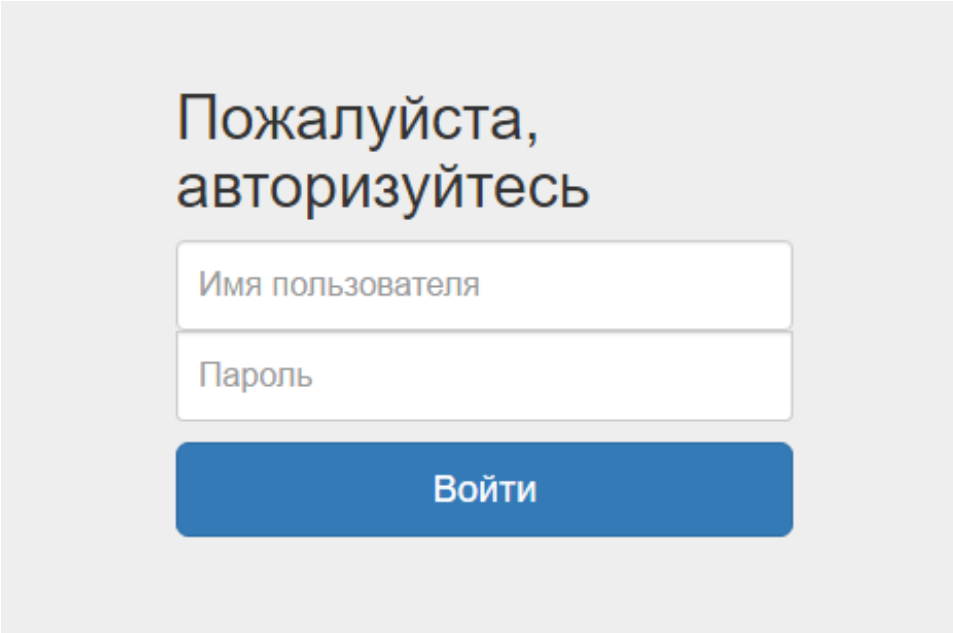
- RSS-подписка на обновления подкастов;
- возможность комментирования подкастов;
- добавление средств контроля знаний студентов;
- сбор статистики.

Можно воспользоваться уже готовыми разработанными модулями, реализующими данные функции, или разработать их самому. Интеграция подобных модулей не повлияет на основную работу системы. Изменение структуры базы

данных будет реализовано средствами стандартной системы миграций в фреймворке Django.

### 2.2.5 Описание реализованной информационной системы

Информационная система требует обязательной авторизации. В зависимости от прав авторизованного пользователя на главной странице будут показаны доступные ему учебные дисциплины.

The image shows a login form on a light gray background. At the top, the text "Пожалуйста, авторизуйтесь" (Please, authorize) is displayed in a large, dark font. Below this text are two white input fields with gray borders. The first field is labeled "Имя пользователя" (Username) and the second is labeled "Пароль" (Password). Below the input fields is a blue button with the text "Войти" (Login) in white.

*Рисунок 13. Окно авторизации информационной системы «Подкаст-терминал образовательного учреждения»*

Преподаватели будут видеть те направления подготовки дисциплины, которые за ними закреплены, а студенты – дисциплины, которые входят в их образовательную программу направления подготовки. Администратор увидит все дисциплины и направления подготовки, которые существуют в системе.

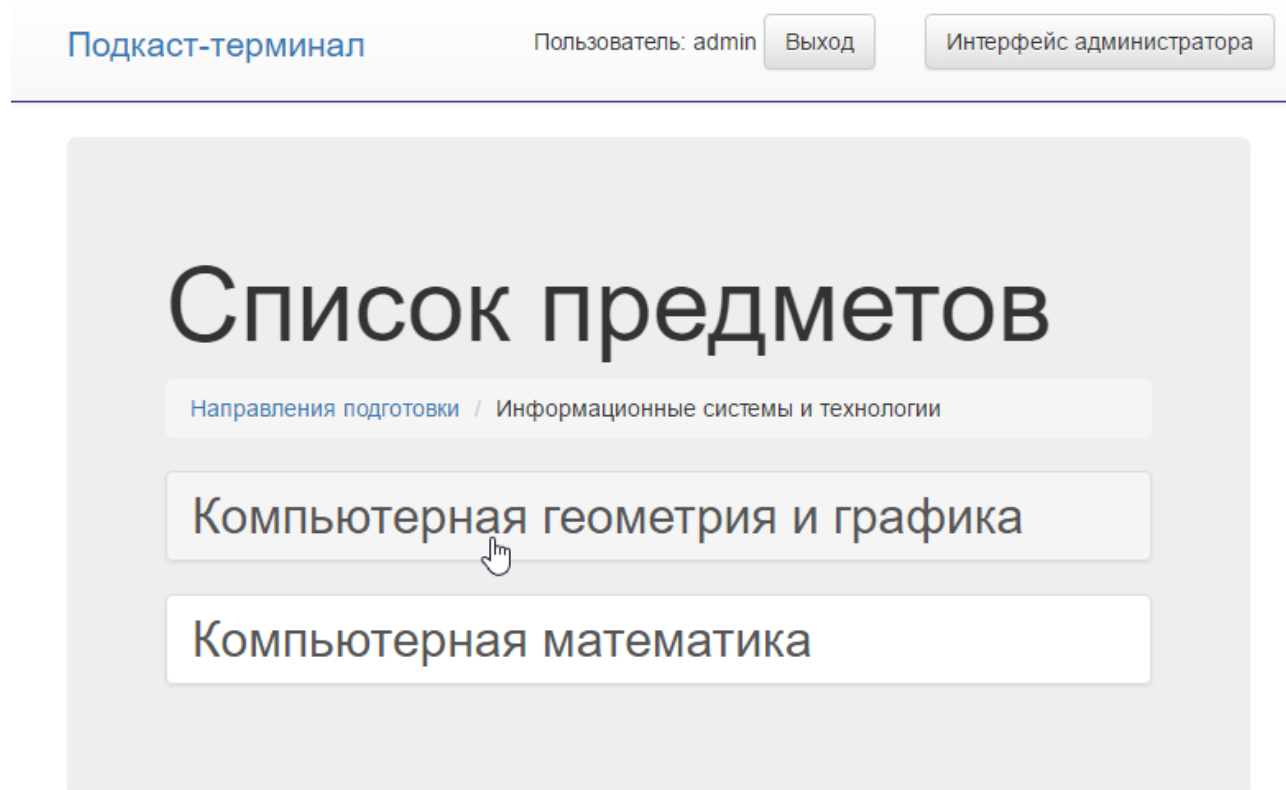


Рисунок 14. Выбор дисциплин в направлении подготовки "Фундаментальная информатика" информационной системы "Подкаст-терминал образовательного учреждения"

В каждое занятие какой-либо дисциплины имеется возможность добавления нескольких подкастов. Такая возможность позволяет разграничивать занятие на несколько заданий или этапов, что позволяет уменьшить объем одного подкаста путем его разбиения на несколько частей. Подкасты будут отображаться по порядку добавления.

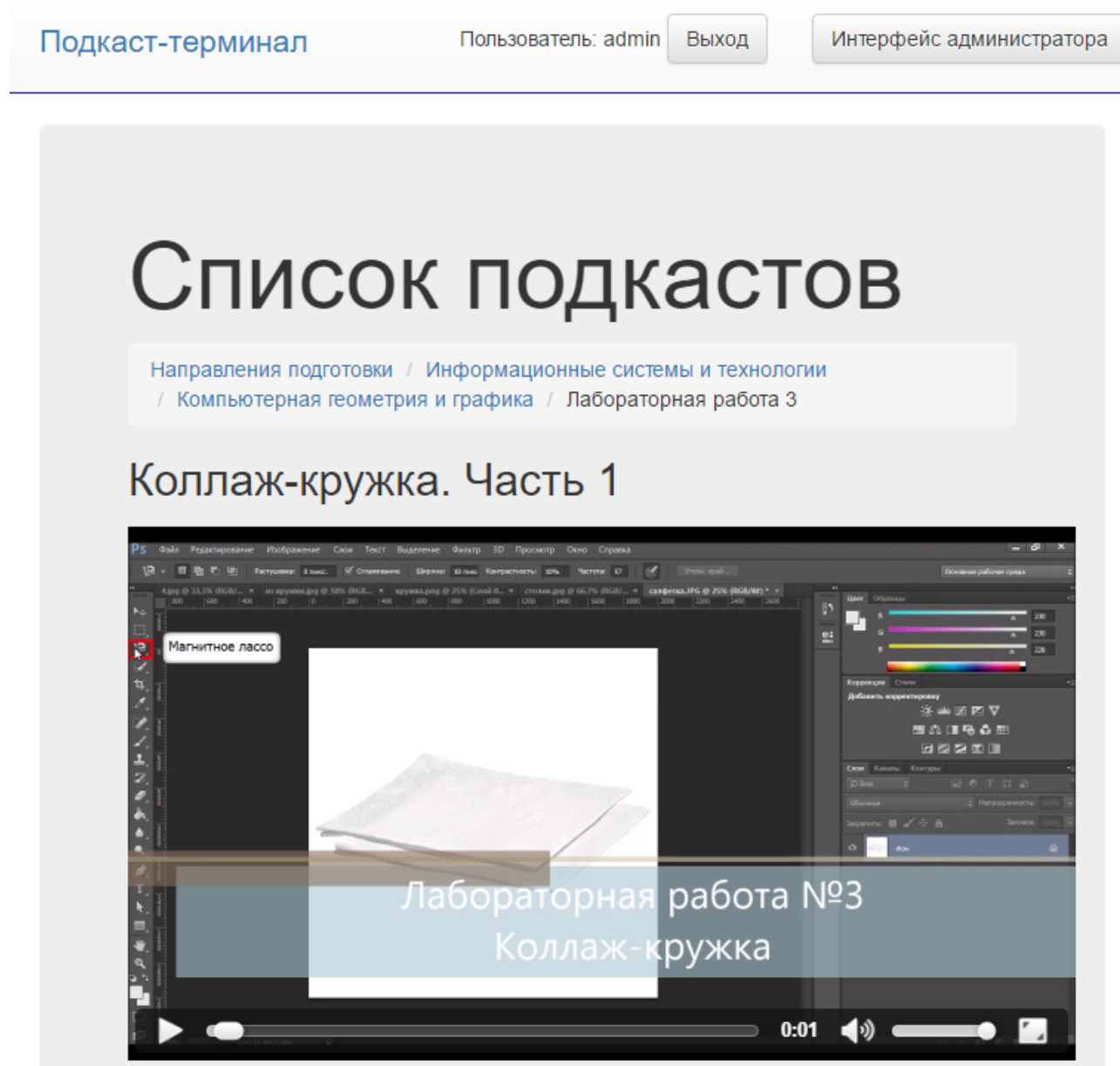


Рисунок 15. Просмотр подкастов первого занятия дисциплины «Компьютерная графика» информационной системы «Подкаст-терминал образовательного учреждения»

Авторизация в административной части информационной системы выглядит также, как и авторизация в саму систему, но студент авторизоваться в ней не сможет.

## Django administration

⊖ Please enter the correct username and password for a staff account. Note that both fields may be case-sensitive.

Username:

Password:

*Рисунок 16. Демонстрация попытки входа студента в интерфейс администратора информационной системы "Подкаст-терминал образовательного учреждения"*

Если администратор или преподаватель уже авторизовались через обычное окно авторизации, то повторный вход не потребуется, и пользователь получит доступ к интерфейсу администратора. В левой части интерфейса располагается перечень модулей, каждый из которых содержит какие-либо сущности. К модулям относятся:

- система аутентификации и авторизации. Управление пользователями и группами;
- модуль управления подкастами;
- структурный модуль управления направлениями подготовки, учебными дисциплинами и занятиями.

В правой части располагается блок с последними выполненными действиями данного пользователя.



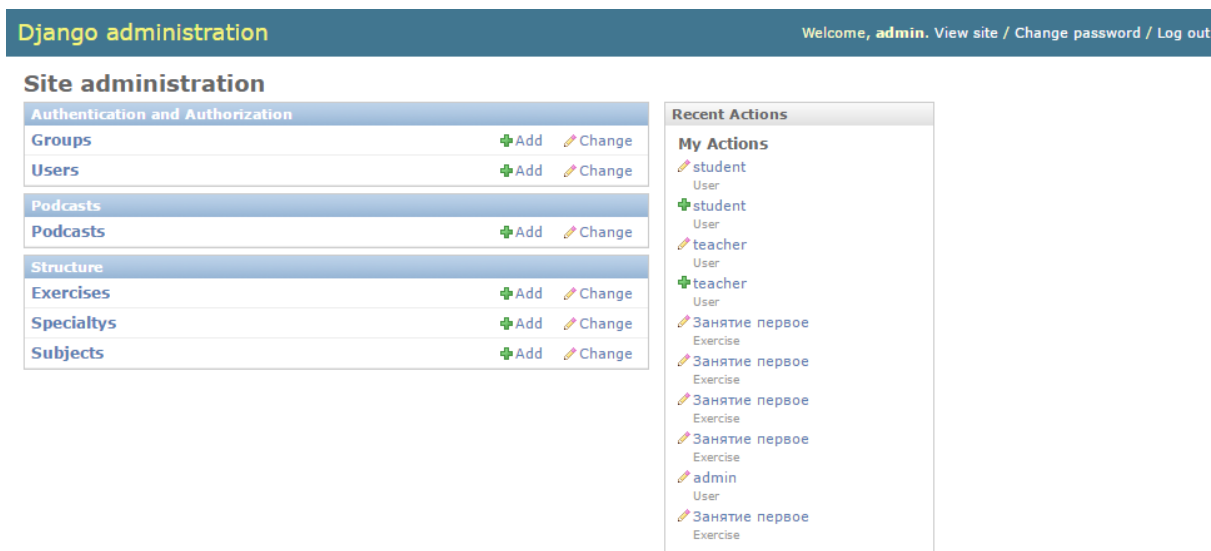


Рисунок 17. Главная страница административной части информационной системы "Подкаст-терминал образовательного учреждения"

Как правило, модули разрабатываются максимально независимыми друг от друга для возможности оперативного подключения модуля в какую-либо другую систему. Каждый модуль содержит перечень сущностей, эквивалентных сущности в базе данных информационной системы. Все сущности из структурного модуля и модуля подкастов имеют связь «один ко многим». Каждая специальность имеет перечень дисциплин. Каждая дисциплина – перечень занятий. Каждое занятие – перечень подкастов.

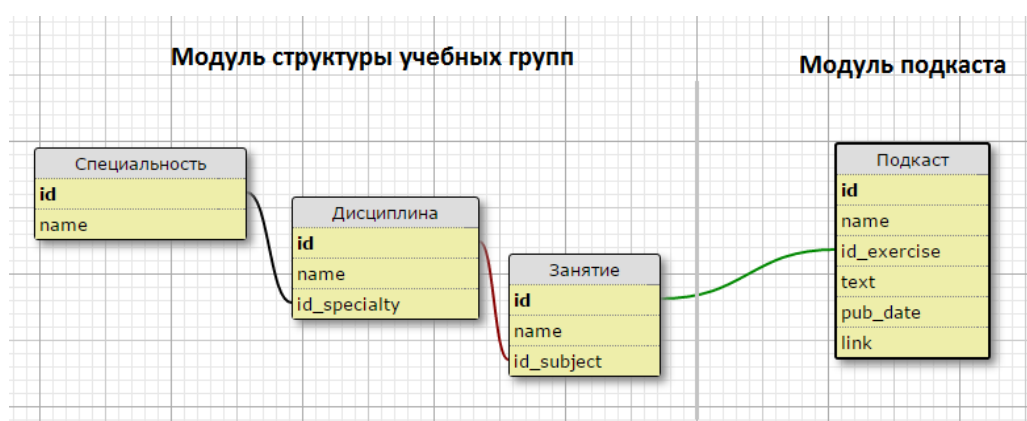


Рисунок 18. Связи сущностей структурного модуля и модуля подкастов в информационной системе «Подкаст-терминал образовательного учреждения»

От лица преподавателя интерфейс предоставляет только те возможности, которые обозначены в его правах доступа. Например, редактирование подка-

стов и занятий в его учебных дисциплинах. Преподаватель не имеет доступа к редактированию системы пользователей, а также к редактированию сущностей направлений подготовки и дисциплин. Имеется возможность добавить данные права в индивидуальном порядке.

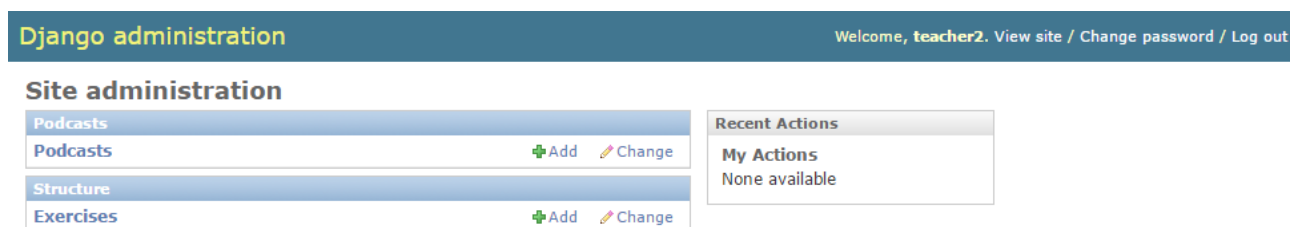


Рисунок 19. Административная часть информационной системы "Подкаст-терминал образовательного учреждения" от лица преподавателя

Право редактировать пользователей и группы имеется только у администратора системы. Администратор может редактировать имя пользователя и его персональную информацию, а также обозначить его статус:

- Является ли пользователь активным. При неактивности пользователь не имеет возможности авторизоваться в информационной системе, не смотря на правильно введенные данные.
- Имеет ли пользователь доступ к интерфейсу администратора. Даже если пользователь имеет доступ к административной части информационной системы, дополнительно пользователь должен иметь права на различные действия с сущностями каких-либо моделей. Права будут рассмотрены далее. Данный статус является активным у преподавателей.
- Является ли пользователь администратором. Данный статус предоставляет пользователю все имеющиеся в системе права и полный доступ к интерфейсу администратора.

Django administration

Welcome, **admin**. [View site](#) / [Change password](#) / [Log out](#)

[Home](#) > [Authentication and Authorization](#) > [Users](#) > [admin](#)

Change user

History

Username:

admin

Required, 30 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

algorithm: pbkdf2\_sha256 iterations: 20000 salt: BhkOk5\*\*\*\*\* hash: 84LgPu\*\*\*\*\*

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using [this form](#).

Personal info

First name:

Сергей

Last name:

Хомутинин

Email address:

huhutymox@gmail.com

Permissions

☒ Active

Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

☒ Staff status

Designates whether the user can log into this admin site.

☒ Superuser status

Designates that this user has all permissions without explicitly assigning them.

Рисунок 20. Редактирование персональной информации пользователя в информационной системе "Подкаст-терминал образовательного учреждения"

Каждый пользователь может принадлежать к каким-либо группам. Группы определяют права входящих в них пользователей и избавляют от необходимости добавлять список прав каждому пользователю индивидуально. Но, при необходимости, это всегда можно сделать. Список прав, по умолчанию, включает в себя по три права для каждой сущности всех моделей системы:

- возможность добавления объекта;
- возможность редактирования объекта;
- возможность удаления объекта.

При необходимости имеется возможность добавления собственных прав для любой из сущностей.

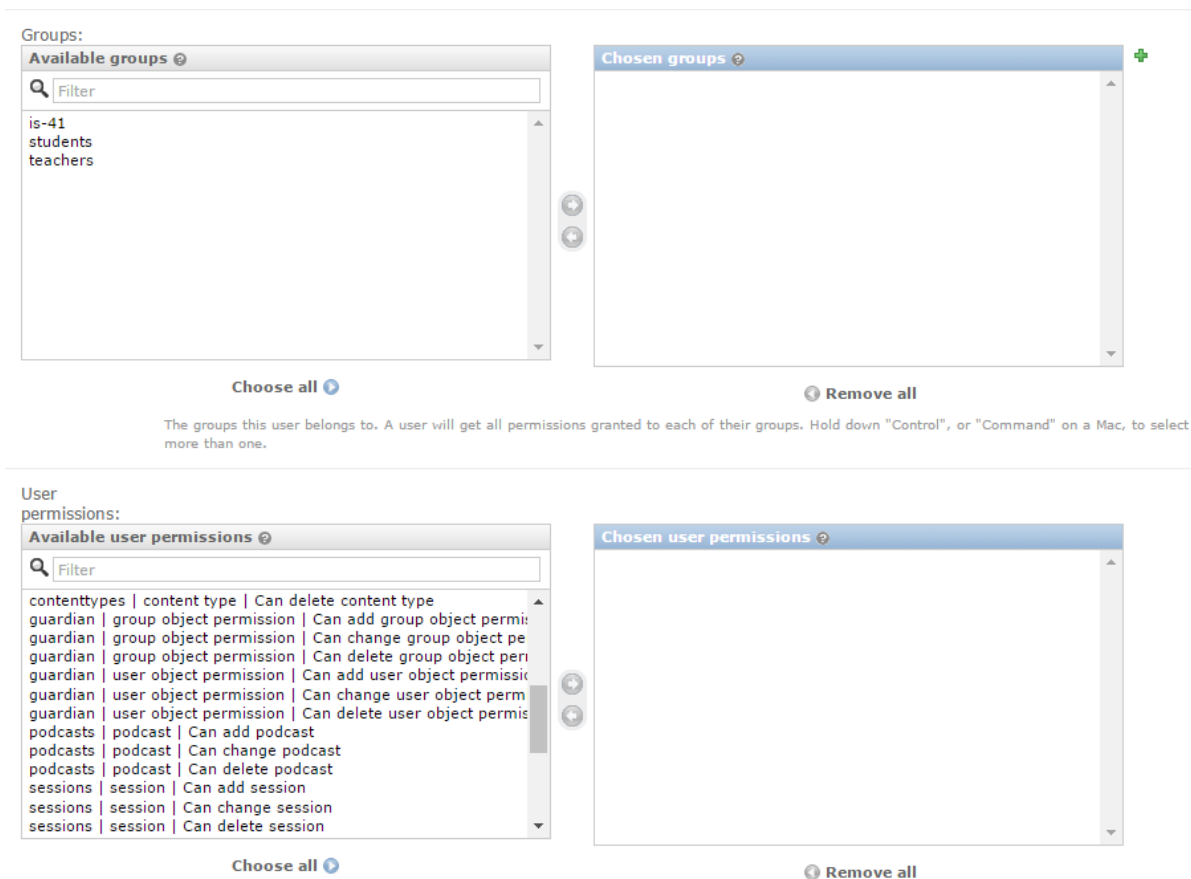


Рисунок 21. Настройка прав пользователя и его принадлежности к группам информационной системы "Подкаст-терминал образовательного учреждения"

Редактирование групп пользователей заключается в добавлении перечня прав. Например, принадлежность пользователя к группе «teachers» автоматически дает ему права на добавление, редактирование и удаление подкастов.

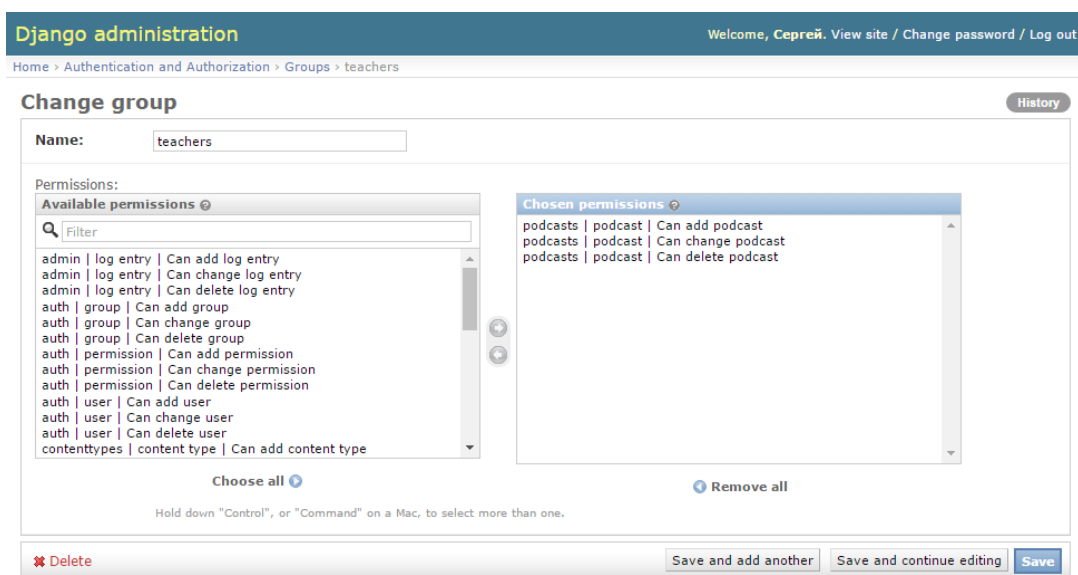


Рисунок 22. Редактирование групп пользователей в информационной системе "Подкаст-терминал образовательного учреждения"

Добавление какой-либо сущности представляет собой заполнение всех полей аналогичной сущности в базе данных информационной системы кроме тех, что имеют какое-либо значение по умолчанию. Поля, связанные с другими сущностями, предлагается выбрать в виде выпадающего списка, либо перейти к добавлению новой записи.

Django administration



Welcome, **Сепрей**. [View site](#) / [Change password](#) / [Log out](#)

[Home](#) > [Structure](#) > [Subjects](#) > Компьютерная графика

### Change subject

[History](#) [View on site](#)

**Subject name:**

**Subject specialty:**   

☒ Delete

*Рисунок 23. Редактирование дисциплины информационной системы «Подкаст-терминал образовательного учреждения»*

Редактирование занятий отлично тем, что имеется возможность быстрого добавления подкастов, принадлежащих этому занятию. Таким образом можно добавить до пяти подкастов в занятие. Если потребуется больше, то придется воспользоваться отдельной формой добавления подкастов в соответствующей модели. Форма редактирования занятия обладает выпадающим списком с выбором родительской дисциплины, также как форма редактирования дисциплины обладает выпадающим списком с выбором направления подготовки. Подкасты в форме редактирования занятия предполагают принадлежность к данному занятию, поэтому аналогичного выпадающего списка здесь нет. В отдельной форме редактирования подкастов, не привязанной к какому-либо занятию такой список имеется.

Django administration
Welcome, Сергей. View site / Change password / Log out

Home > Structure > Exercises > Занятие первое

### Change exercise

History
View on site

Exercise name:
Занятие первое

Exercise subject:
Компьютерная графика

#### Podcasts

Podcast: Задание 1. Малина
Delete

Podcast name:
Задание 1. Малина

Podcast text:
Выделение фрагментов изображений

Date published:
Date: 2016-05-14 Today
Time: 22:17:51 Now

Link:
Currently: ./Малина.mp4
Change: Выберите файл Файл не выбран

Podcast: Задание 2. Выделение банана
Delete

Podcast name:
Задание 2. Выделение банана

Podcast text:
Выделение фрагментов изображений

Date published:
Date: 2016-05-14 Today
Time: 22:18:02 Now

Link:
Currently: ./Неспелый\_банан.mp4
Change: Выберите файл Файл не выбран

Podcast: #3

Рисунок 24. Редактирование занятия информационной системы  
"Подкаст-терминал образовательного учреждения"

Таким образом, в рамках данной работы была поставлена задача разработать локальный прототип подкаст-терминала образовательного учреждения, и данная задача включает в себя не все этапы реализации информационной системы.

46

Общее соотношение реализованных и описанных теоретически этапов представлено на следующей таблице:

*Таблица 2.  
Реализация и теоретическое описание  
этапов разработки информационной системы*

	1 этап	2 этап	3 этап	4 этап	5 этап
<b>Реализовано</b>	Да	Да	Нет	Нет	Да
<b>Описано теоретически</b>	Да	Да	Да	Да	Да

Следовательно, в данной работе имеется теоретическое описание всех этапов разработки системы. Результатом разработки является локальный прототип информационной системы «Подкаст-терминал образовательного учреждения» с реализованной сопроводительной документацией в виде руководства пользователя. Развертывание системы на сервере и интеграция дополнительных модулей не производилась.

### **2.3 Результаты апробации, техническая документация**

Техническая документация представлена руководством по эксплуатации системы для всех типов пользователей – администратора, преподавателя и студента (См. Приложение 1 и файл Руководство.pdf).

Руководство пользователя имеет следующую структуру:

- комплекс программных и аппаратных требований;
- функции различных типов пользователей;
- интерфейс администратора;
- интерфейс преподавателя;
- интерфейс студента.

Результаты работы представлены в межвузовских сборниках научных работ 2015 и 2016 годов выпуска:

1. Хомутинин С.С., Сардак Л.В. Особенности разработки подкаст-терминала учебного назначения // Актуальные вопросы преподавания математики, информатики и информационных технологий: межвузовский сборник научных работ. Екатеринбург: Урал. гос. пед. ун-т, 2016.
2. Хомутинин С.С., Сардак Л.В. Технологии развертывания подкаст-терминала в образовательном учреждении // Актуальные вопросы преподавания математики, информатики и информационных технологий: межвузовский сборник научных работ. Екатеринбург: Урал. гос. пед. ун-т, 2015.

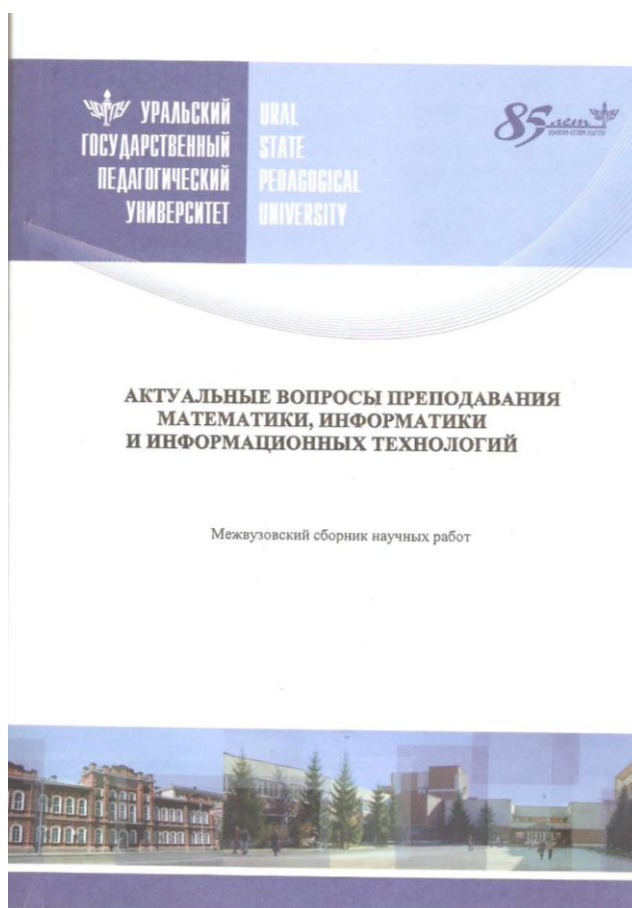


Рисунок 25. Обложка сборника "Актуальные вопросы преподавания математики, информатики и информационных технологий"

Подана заявка на регистрацию программного продукта.



## **Заключение**

Результатом выпускной квалификационной работы является разработка локального прототипа информационной системы «Подкаст-терминал образовательного учреждения» в соответствии с техническим заданием. Система организует систему закрытого подкастинга в образовательном учреждении. В процессе разработки были решены следующие задачи:

1. Произведен анализ технологий реализации подкаст-терминала, выявлена специфика подкаст-терминала образовательного учреждения.
2. Проанализирован и обоснован выбор аппаратно-программной платформы, определена структура, объекты web-интерфейса подкаст-терминала образовательного учреждения.
3. В соответствии с техническим заданием смоделирован, спроектирован и реализован локальный прототип информационной системы «Подкаст-терминал образовательного учреждения».
4. Подготовлена техническая документация, проведена апробация.

Разработанная информационная система соответствует техническому заданию.

## Список информационных источников

1. Абдуразакова Д.С., Демашина В.Е., Лучанинов Д.В. Анализ сравнения систем управления обучением Joomla и Wordpress с точки зрения разработки web-сайтов образовательного назначения // Постулат. Электронный научный журнал. 2015. №2.
2. Бесплатное обучение веб-разработке // codenamecrud.ru URL: <http://codenamecrud.ru/pages/curriculum>
3. Куликович Т. О. Основы научного цитирования. Минск: 2010.
4. Лутц М. Изучаем Python. - 4 изд. Символ-Плюс, 2010.
5. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 3 изд. СПб: Питер, 2015.
6. Рейтинг CMS // ratingruneta.ru URL: <http://www.ratingruneta.ru/cms/>
7. Система управления содержимым // ru.wikipedia.org URL: [https://ru.wikipedia.org/wiki/Система\\_управления\\_содержимым](https://ru.wikipedia.org/wiki/Система_управления_содержимым)
8. Теория и практика звукозаписи: Сам себе подкаст-терминал // tipz.umputun.com URL: <http://tipz.umputun.com/2008/08/blog-post.html>
9. Федеральный государственный образовательный стандарт высшего образования от 12 марта 2015 г. №219 // Консультант плюс. в ред. Приказа Минобрнауки России от 09.09.2015 № 999.
10. Фреймворк - Википедия // ru.wikipedia.org URL: <https://ru.wikipedia.org/wiki/Фреймворк>
11. Хомутинин С.С., Сардак Л.В. Особенности разработки подкаст-терминала учебного назначения // Актуальные вопросы преподавания математики, информатики и информационных технологий: межвузовский сборник научных работ. Екатеринбург: Урал. гос. пед. ун-т, 2016.
12. Хомутинин С.С., Сардак Л.В. Технологии развертывания подкаст-терминала в образовательном учреждении // Актуальные вопросы преподавания мате-

- матики, информатики и информационных технологий: межвузовский сборник научных работ. Екатеринбург: Урал. гос. пед. ун-т, 2015.
- 13.Чугреев В. Л. Особенности реализации MVC-архитектуры в веб-приложениях // Молодой ученый. — 2015. — №7. — С. 66-71.
  - 14.Apple - iTunes - Подкасты // Apple.com URL:  
<http://www.apple.com/ru/itunes/podcasts/fanfaq.html>
  - 15.Django documentation // djangoproject.com URL:  
<https://docs.djangoproject.com/en/1.9/>
  - 16.Django на русском // djbook.ru URL: <http://djbook.ru/>
  - 17.iTunes // apple.com URL: <http://www.apple.com/ru/itunes/>
  - 18.PyCharm // JetBrains.com URL: <https://www.jetbrains.com/pycharm/>
  - 19.Oxford Dictionaries // oxforddictionaries.com URL:  
<http://www.oxforddictionaries.com/definition/english/podcast>
  - 20.Podcast // en.wikipedia.org URL: <http://en.wikipedia.org/wiki/Podcast>
  - 21.PostgreSQL | Наиболее продвинутая открытая СУБД в мире // postgresql.ru.net URL: <http://postgresql.ru.net/>
  - 22.The new Oxford American dictionary / ed. by Erin McKean. Second edition. Oxford University Press, 2005. – 2051 p.
  - 23.Welcome to Python.org // Python.org URL: <https://www.python.org/>

## **Приложения**

### **Приложение 1.**

ФГБОУ ВО «Уральский государственный педагогический университет»

Институт математики, информатики и информационных технологий

Кафедра информационно-коммуникационных технологий в образовании

Автор: Хомутинин С. С.

# **Подкаст-терминал учебного назначения**

**Руководство пользователя**

Екатеринбург 2016 г.

## **Введение**

Подкаст-терминал учебного назначения представляет собой онлайн хранилище учебных медиафайлов – подкастов. Доступ к подкастам определяется правами каждого конкретного пользователя. Сервис предполагает несколько типов пользователей с различными правами и обязанностями. Подкасты располагаются в определенной структуре (направление подготовки – дисциплина – занятие - подкаст). Доступ определенных пользователей к элементам структуры устанавливается в индивидуальном порядке. Такое структурирование позволяет удобно делиться и просматривать учебный контент. В руководстве рассмотрен комплекс требований к системе и ее компонентам, а также права и основные функции каждого типа пользователей системы.

## **Комплекс требований**

Аппаратные требования к серверу:

- процессор x64 с тактовой частотой 1.4 ГГц;
- 2 ГБ оперативной памяти;
- 250Гб объем жесткого диска.

Программные требования к серверу:

- любая 64-битная Linux-подобная операционная система.

Программные требования к персональному компьютеру клиента:

- наличие любого современного веб-браузера.

Требования к записи контента:

- наличие хорошего микрофона;
- возможность записи видео в разрешении минимум 420p;
- веб-камера при необходимости;
- невысокий хронометраж (10-20 мин.).

## Функции различных типов пользователей

Подкаст-терминал направлен на организацию доступа к учебным материалам студентам. Владеют учебными материалами, обычно, преподаватели. За структуру подкастов и систему пользователей отвечает администратор подкаст-терминала. Итого, имеется три типа пользователей:

- администратор;
- преподаватель;
- студент.

## Интерфейс администратора

Администратор задает первоначальную структуру, управляет пользователями и распределяет права доступа. Получить доступ к данному интерфейсу может как администратор, так и преподаватель, но возможности будут различаться.

Работа в интерфейсе администратора представляет собой редактирование сущностей различных моделей. В основном, редактирование любой сущности выглядит одинаково. При выборе сущности открывается список уже существующих экземпляров объекта. Кнопка «Add XXX» позволяет добавить новый экземпляр, где XXX – имя выбранной сущности. Выбор экземпляра перенаправляет на страницу с его редактированием. После внесения изменений необходимо нажать кнопку «Save». Список полей у сущностей различный, рассмотрим существующие в системе сущности подробнее.

**Users** – сущность, определяющая какого-либо пользователя. Список полей данной сущности:

- Username – логин пользователя;
- Password – пароль пользователя. Зашифрован и недоступен для редактирования;
- First name – имя пользователя;
- Last name – фамилия пользователя;

- Email address – электронная почта;
- Active – активность пользователя. Выключение данного статуса делает пользователя неактивным, блокируя ему авторизацию;
- Staff status – доступность пользователю интерфейса администратора. Включение данной функции разрешает только войти в интерфейс, но доступ к редактированию различных сущностей определяется правами пользователя (см. далее);
- Superuser status – статус суперпользователя. Дает пользователю все имеющиеся права в системе и принадлежность ко всем группам;
- Groups – перечень групп, к которым принадлежит пользователь;
- User permissions – индивидуальные права пользователя;
- Last login – дата последнего входа в систему;
- Date joined – дата регистрации пользователя.

**Groups** – сущность групп пользователей. Поля данной сущности:

- Name – название группы;
- Permissions – перечень прав данной группы. Пользователи, принадлежащие группе, автоматически получают перечень прав этой группе.

**Specialtys** – сущность, определяющая специальность. Содержит одно поле – Specialty name – название данной специальности.

**Subjects** – сущность, определяющая учебную дисциплину. Поля следующие:

- Subject name – название дисциплины;
- Subject specialty – специальность, которой принадлежит данная дисциплина.

**Exercises** – сущность, определяющая занятие. Содержит следующие поля:

- Exercise name – название занятия;
- Exercise subject – дисциплина, которой принадлежит данное занятие.



В рамках данной сущности имеется возможность добавления до пяти подкастов:

**Podcasts** – сущность подкаста. Имеется возможность добавления как из интерфейса добавления занятия, так и отдельно. Поля следующие:

- Podcast name – название подкаста;
- Podcast exercise – занятие, которому принадлежит данный подкаст. Если добавление подкаста производится через интерфейс добавления занятия, данное поле отсутствует и по умолчанию равно выбранному занятию;
- Podcast text – описание подкаста, либо сопроводительный текст;
- Date published – дата и время публикации подкаста;
- Link – ссылка на видеофайл подкаста.

### **Интерфейс преподавателя**

Преподаватель осуществляет свою работу средствами интерфейса администратора. Доступ определяется его правами и обычно соответствует какому-либо перечню экземпляров сущностей занятий (в принадлежащей ему дисциплине) и подкастов. Поля сущностей Exercises и Podcasts описаны в руководстве по интерфейсу администратора.

### **Интерфейс студента**

Студент имеет доступ к перечню дисциплин своего направления подготовки и занятий, принадлежащих данным дисциплинам. Студенту может выбирать доступные ему разделы по следующим уровням вложенности:

1. дисциплина;
2. занятие;
3. подкаст.

Выбирать направление подготовки студент не может. Воспроизведение подкаста осуществляется средствами стандартного видеоплеера HTML5. Имеется возможность остановить видеопоток, выбрать необходимый временной промежуток, изменить громкость или развернуть видео на весь экран.

## **Заключение**

В данном руководстве были рассмотрены основные функции информационной системы «Подкаст-терминал образовательного учреждения» со стороны администратора, преподавателя и студента. При возникновении дополнительных вопросов необходимо обратиться к администратору системы.